2018

Audio-visual Video Recognition Through Super Descriptor Tensor Decomposition and Low-rank and Sparse Representation

Muhammad Rizwan Khokher
University of Wollongong

UNIVERSITY OF WOLLONGONG
COPYRIGHT WARNING

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site. You are reminded of the following:

This work is copyright. Apart from any use permitted under the Copyright Act 1968, no part of this work may be reproduced by any process, nor may any other exclusive right be exercised, without the permission of the author.

Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

Recommended Citation

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au
Audio-visual Video Recognition Through Super Descriptor Tensor Decomposition and Low-rank and Sparse Representation

A thesis submitted in partial fulfillment of the requirements for the award of the degree

Doctor of Philosophy

by

Muhammad Rizwan Khokher

School of Electrical, Computer, and Telecommunications Engineering

UNIVERSITY OF WOLLONGONG

October 2018
Statement of Originality

I, Muhammad Rizwan Khokher, declare that this thesis, submitted in partial fulfillment of the requirements for the award of Doctor of Philosophy, in the School of Electrical, Computer, and Telecommunications Engineering, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications at any other academic institution.

Muhammad Rizwan Khokher
October 2018
Contents

Acronyms .................................................. XII
Abstract ................................................... XIV
Acknowledgments ........................................ XVI

1 Introduction ................................................. 1
   1.1 Thesis Significance .................................... 2
   1.2 Research Objectives ..................................... 3
   1.3 Thesis Contributions .................................... 3
   1.4 Thesis Organization .................................... 4
   1.5 Research Publications .................................. 5

2 Review of Audio-visual Video Recognition ................. 7
   2.1 Introduction ........................................... 8
   2.2 Audio Feature Extraction ............................... 8
      2.2.1 Audio Attributes .................................. 9
      2.2.2 Audio Feature Descriptors ......................... 10
         2.2.2.1 Temporal Domain ............................... 10
         2.2.2.2 Frequency Domain ............................. 10
         2.2.2.3 Cepstral Domain ............................... 11
   2.3 Visual Feature Extraction .............................. 12
      2.3.1 Spatio-temporal Interest Point Detection ........ 12
         2.3.1.1 Spatio-temporal Corner Detectors ............ 12
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.1.2</td>
<td>Spatio-temporal Filtering Methods</td>
<td>13</td>
</tr>
<tr>
<td>2.3.1.3</td>
<td>Global Information based Techniques</td>
<td>13</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Trajectory Formation</td>
<td>14</td>
</tr>
<tr>
<td>2.3.3</td>
<td>Visual Feature Descriptors</td>
<td>15</td>
</tr>
<tr>
<td>2.3.3.1</td>
<td>Appearance based Visual Descriptors</td>
<td>15</td>
</tr>
<tr>
<td>2.3.3.2</td>
<td>Motion based Visual Descriptors</td>
<td>16</td>
</tr>
<tr>
<td>2.4</td>
<td>Global Feature Representation</td>
<td>17</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Vocabulary Generation</td>
<td>18</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Local Feature Encoding</td>
<td>19</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Pooling of Encoded Features</td>
<td>21</td>
</tr>
<tr>
<td>2.5</td>
<td>Video Classification</td>
<td>22</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Instance based Learning</td>
<td>23</td>
</tr>
<tr>
<td>2.5.2</td>
<td>Logic based Learning</td>
<td>24</td>
</tr>
<tr>
<td>2.5.3</td>
<td>Statistical and Graphical Approaches</td>
<td>24</td>
</tr>
<tr>
<td>2.5.4</td>
<td>Support Vector Machines</td>
<td>25</td>
</tr>
<tr>
<td>2.5.5</td>
<td>Neural Networks and Deep Learning</td>
<td>25</td>
</tr>
<tr>
<td>2.6</td>
<td>Chapter Summary</td>
<td>28</td>
</tr>
</tbody>
</table>

3 Visual Feature Extraction | 29 |
| 3.1     | Introduction | 29 |
| 3.2     | Refined Dense Trajectories | 31 |
| 3.2.1   | Region of Interest based Sampling | 31 |
| 3.2.2   | Short-window Video Stabilization | 32 |
| 3.2.3   | Tracking of Interest Points | 32 |
| 3.2.4   | Descriptor Computation | 33 |
| 3.2.5   | Experimental Results and Analysis of the RDT | 34 |
| 3.2.5.1 | Datasets and Evaluation Protocol | 34 |
| 3.2.5.2 | Experimental Method | 37 |
| 3.2.5.3 | Effects of Trajectory Length | 37 |
| 3.2.5.4 | Visual Analysis | 38 |
| 3.2.5.5 | Evaluation based on Classification Accuracy and Computation Time | 39 |
3.3 Spatio-temporal Interest Point Detector based on Low-rank and Group-sparse Matrix Approximation

3.3.1 LRGS Representation of Interest Points

3.3.2 Approximation of LRGS Components

3.3.3 Extraction of Desired STIPs

3.3.4 Experimental Results and Analysis of the LRGS-STIP

3.3.4.1 Datasets and Evaluation Protocol

3.3.4.2 Experimental Method

3.3.4.3 Evaluation based on Detection Ratio

3.3.4.4 Evaluation based on Classification Accuracy

3.4 Chapter Summary

4 Super Descriptor Tensor Decomposition

4.1 Introduction

4.2 Encoding Local Feature Descriptors

4.2.1 Gaussian Mixture Modeling

4.2.2 Sparse Dictionary Learning

4.2.3 Feature Encoding

4.3 Tensor Decomposition and Feature Selection

4.3.1 TUCKER-3 Tensor Decomposition

4.3.2 Higher-order Orthogonal Interactions

4.3.3 Fisher Ranking for Feature Selection

4.4 Experimental Results and Analysis

4.4.1 Datasets and Evaluation Protocol

4.4.2 Experimental Setup

4.4.3 Performance of Different Classifiers

4.4.4 Effects of Dictionary Size in Feature Encoding

4.4.5 Evaluation of Tensor Decomposition Algorithms

4.4.6 Effects of Fitness Threshold $\theta$ during Tensor Decomposition

4.4.7 Analysis of Feature Selection after Tensor Decomposition

4.4.8 Analysis of Different Configurations of the SDTD Model

IV
4.4.9 Comparison of the SDTD model with other Feature Representation Models .............................................. 69
  4.4.9.1 Implementation of Different Feature Representation Models ..................................................... 69
  4.4.9.2 Comparison in terms of Classification Accuracy and Dimensionality ...................................... 69
  4.4.9.3 Comparison in terms of Computation Time ................................................................. 71

4.5 Chapter Summary ................................................................. 72

5 Applications for Visual Video Recognition .......................... 73
  5.1 Introduction ................................................................. 73
  5.2 Dynamic Scene Recognition ............................................. 74
    5.2.1 Related Work ......................................................... 74
    5.2.2 Experimental Method .............................................. 76
    5.2.3 Classification Results for Dynamic Scene Recognition ...................................................... 77
    5.2.4 Comparison with State-of-the-art Methods of Dynamic Scene Recognition ............................... 79
  5.3 Action Recognition ......................................................... 81
    5.3.1 Related Work ......................................................... 81
    5.3.2 Experimental Method .............................................. 84
    5.3.3 Classification results for Action Recognition ........................................................................ 85
    5.3.4 Comparison with State-of-the-art Methods of Action Recognition ............................................ 87
  5.4 Chapter Summary ................................................................. 90

6 Applications for Audio-visual Video Recognition .................... 91
  6.1 Introduction ................................................................. 91
  6.2 Human Interaction Recognition ......................................... 92
    6.2.1 Related Work ......................................................... 92
    6.2.2 Experimental Method .............................................. 94
    6.2.3 Classification Results for Human Interaction Recognition .................................................... 94
    6.2.4 Comparison with State-of-the-art Methods for Human Interaction Recognition ........................ 95
List of Figures

3.1 Illustration of the trajectory and descriptor computation, adapted from [23]. (a) A hand waving scene. Red points show the interest points to be tracked and green tracks represent the trajectories. A median filter is applied to the dense optical flow field to track the points. (b) Descriptors such as HOG and MBH are computed along the trajectories within a volume of size $R \times R \times L$, which is subdivided into cells of size $r_x \times r_y \times \ell$. ........................................... 33

3.2 Sample video frames from Maryland “in-the-wild” [103] and YUP-PEN dynamic scenes [104] datasets. ................................. 36

3.3 Classification rate versus trajectory length $L$ for Maryland and YUP-PEN datasets. ............................................................. 38

3.4 Refined dense trajectories: (a) (d) A waterfall and windmill scene from YUPPEN dynamic scenes dataset [104]; (b) (e) Trajectories computed by the dense trajectories method in [23], the green lines represent the motion trajectories, and the red dots represent the end points of the trajectories; (c) (f) The proposed refined dense trajectories, the irrelevant trajectories (red dots in static textured region in (b) and (e)) have been removed, and only trajectories in the ROI are kept. .......................................................... 39
3.5 STIP detection through the proposed LRGS-STIP detector. Input video shows a scene of a moving train. The SIPs are detected using FAST corners, SURF features, and Canny edges. The desired STIPs are then detected via the proposed LRGS-STIP detector. ............ 45

3.6 Some sample video frames from action recognition datasets: KTH [110], UCF [111], YouTube [112], and MSR-I [113]. ............ 47

4.1 Block diagram of the proposed SDTD method. First, the individual tensors obtained after SDV coding are concatenated to get tensors $X$ and $X^t$ for training and test datasets, respectively. The training tensor $X$ is decomposed through the TUCKER-3 tensor decomposition using the HOOI algorithm. Training features are obtained from the core tensor $G$ after the decomposition. The orthogonal basis factors $U$ are used to get the test features from $G^t = X^t \times (U^T)$. The feature selection is performed using Fisher ranking to select discriminative features. Finally, a classifier is used to classify the video segments. .................. 58

4.2 Sample video frames from THIVD [129] and Parliament [130] datasets. 61

4.3 Classification rate versus dictionary size for SDV coding in the SDTD model. .................. 64

4.4 Classification rate versus number of features using Fisher ranking, Student’s t-test, and mutual information for Maryland, YUPPEN, TVHID, and Parliament datasets. .................. 67

4.5 Classification rate versus number of features of different configurations within the SDTD model for Maryland, YUPPEN, TVHID, and Parliament datasets. .................. 68

5.1 Classification rate versus number of features for KTH, UCF, and YouTube datasets. .................. 85

6.1 Sample video frames from the MediaEval VSD2014 dataset [171]. 99

6.2 MAP2014 score versus number of features for the Test (Hollywood) and Generalization (YouTube) subsets. .................. 101
List of Tables

2.1 Local descriptors for audio feature extraction .................. 12
2.2 Spatio-temporal interest point detectors for videos ............. 14
2.3 Various approaches for trajectory formation in videos .......... 15
2.4 Visual feature descriptors for videos .......................... 18
2.5 Vocabulary generation methods for global feature representation. 19
2.6 Representative feature encoding techniques based on combination of codewords .............................................. 21
2.7 Representative feature encoding techniques based on difference between features and codewords .............................. 22
2.8 Machine learning methods for video classification .............. 27

3.1 Number of trajectories, total computation time (trajectories and descriptors), and CR ± std (average and per-category) of the proposed RDT and the method of Wang et al. [23], for Maryland “in-the-wild” dataset. For a better comparison between the two methods, the global motion compensation using short-window video stabilization is not added .............................................. 40
3.2 Number of trajectories, total computation time (trajectories and descriptors), and CR ± std (average and per-category) of the proposed RDT and method of Wang et al. [23], for YUPPEN dynamic scenes dataset ................................................................. 41
3.3 Average ratio ± std (in percent) of valid STIPs detected by the proposed LRGS-STIP and other detectors for MSR-I dataset. The ratio is the number of STIPs detected for actions divided by the total number of detected STIPs. 

3.4 Average CR ± std (in percent) of the proposed LRGS-STIP and other STIP detectors for KTH, UCF, and YouTube datasets.

3.5 Average CR ± std (in percent) of the proposed LRGS-STIP and other STIP detectors for UCF and YouTube datasets. The global motion compensation is added for all the detectors.

4.1 Average CR ± std for the proposed SDTD method using naive Bayes, kNN, non-linear SVM, linear SVM, and extreme learning machines classifiers.

4.2 Average CR ± std for the SDTD model using HODA and HOOI algorithms.

4.3 Average CR ± std (in percent) for the SDTD model using different fitness thresholds $\theta$ (without feature selection).

4.4 Average CR ± std of the SDTD method and four different feature representation methods: SPM, LLC, FV, and SDV.

4.5 Average CR ± std of the SDTD, FV, and SDV methods for the same number of features ($f$).

4.6 Run-time (in seconds) of different global feature representation methods: SDTD, SPM, LLC, FV, and SDV, for feature extraction, training, and testing on Maryland dataset.

5.1 Confusion matrices of the proposed visual recognition system for Maryland and YUPPEN dataset.

5.2 Average CR ± std (in percent) of the proposed RDT+SDTD and other methods for Maryland and YUPPEN datasets.

5.3 The Friedman’s test $p$-value for the proposed RDT+SDTD in comparison with other methods for Maryland dataset.

5.4 The Friedman’s test $p$-value for the proposed RDT+SDTD in comparison with other methods for YUPPEN dataset.
5.5 Confusion matrices of the proposed visual recognition system for action recognition using KTH, UCF, and YouTube datasets. 86
5.6 Average CR ± std (in percent) of the proposed LRGS-STIP+SDTD and other methods for KTH, UCF, and YouTube datasets. 87
5.7 The Friedman’s test p-value for the proposed LRGS-STIP+SDTD in comparison with other methods for KTH dataset. 88
5.8 The Friedman’s test p-value for the proposed LRGS-STIP+SDTD in comparison with other methods for UCF dataset. 89
6.1 Confusion matrices of the proposed audio-visual recognition system for human interaction recognition datasets: TVHID and Parliament. 95
6.2 Average CR ± std (in percent) of the proposed LRGS-STIP+SDTD and other methods for TVHID and Parliament datasets. 96
6.3 The Friedman’s test p-value for the proposed LRGS-STIP+SDTD in comparison with other methods for TVHID dataset. 96
6.4 The Friedman’s test p-value for the proposed LRGS-STIP+SDTD in comparison with other methods for Parliament dataset. 97
6.5 MAP2014 scores ± std (in percent) of the RDT+SDTD method and the VSD2014 participating teams for the Test (Hollywood) and Generalization (YouTube) subsets. 102
<table>
<thead>
<tr>
<th>Acronyms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>Two dimensional</td>
</tr>
<tr>
<td>3D</td>
<td>Three dimensional</td>
</tr>
<tr>
<td>BoW</td>
<td>Bag-of-words</td>
</tr>
<tr>
<td>CR</td>
<td>Classification rate</td>
</tr>
<tr>
<td>ELM</td>
<td>Extreme learning machines</td>
</tr>
<tr>
<td>FAST</td>
<td>Features from accelerated segment test</td>
</tr>
<tr>
<td>FV</td>
<td>Fisher vector</td>
</tr>
<tr>
<td>GMM</td>
<td>Gaussian mixture model</td>
</tr>
<tr>
<td>HODA</td>
<td>Higher-order discriminant analysis</td>
</tr>
<tr>
<td>HOG</td>
<td>Histogram of oriented gradients</td>
</tr>
<tr>
<td>HOOI</td>
<td>Higher-order orthogonal interactions</td>
</tr>
<tr>
<td>kNN</td>
<td>$k$-nearest neighbors</td>
</tr>
<tr>
<td>LLC</td>
<td>Locality-constrained linear coding</td>
</tr>
<tr>
<td>LRGS</td>
<td>Low-rank and group-sparse</td>
</tr>
<tr>
<td>MAP</td>
<td>Mean average precision</td>
</tr>
<tr>
<td>MBH</td>
<td>Motion boundary histogram</td>
</tr>
<tr>
<td>Acronyms</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>MFCC</td>
<td>Mel-frequency cepstral coefficients</td>
</tr>
<tr>
<td>NB</td>
<td>Naive Bayes</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial basis function</td>
</tr>
<tr>
<td>RDT</td>
<td>Refined dense trajectories</td>
</tr>
<tr>
<td>ROI</td>
<td>Region of interest</td>
</tr>
<tr>
<td>SDV</td>
<td>Super descriptor vector</td>
</tr>
<tr>
<td>SDTD</td>
<td>Super descriptor tensor decomposition</td>
</tr>
<tr>
<td>SMOTE</td>
<td>Synthetic minority over-sampling technique</td>
</tr>
<tr>
<td>SPM</td>
<td>Spatial pyramid matching</td>
</tr>
<tr>
<td>STIP</td>
<td>Spatio-temporal interest point</td>
</tr>
<tr>
<td>Std</td>
<td>Standard deviation</td>
</tr>
<tr>
<td>SURF</td>
<td>Speeded-up robust features</td>
</tr>
<tr>
<td>SVM</td>
<td>Support vector machines</td>
</tr>
<tr>
<td>SWVS</td>
<td>Short-window video stabilization</td>
</tr>
<tr>
<td>TD</td>
<td>Tensor decomposition</td>
</tr>
</tbody>
</table>
Abstract

This work deals with audio-visual video recognition using machine learning. A general audio-visual video recognition system first extracts auditory and visual feature descriptors, then represents the extracted bi-modal features using feature encoding techniques, and finally performs recognition using a machine learning classifier. This work adapts a similar pipe-line, contributing to the first two major components: visual feature extraction and global feature representation.

Visual feature extraction is a vital step in video recognition. In general, the visual feature extraction starts by detecting spatio-temporal interest points where the features are most discriminative in a video. There are a few problems associated with existing spatio-temporal interest point detectors. Firstly, the detectors are either too sparse, which leads to loss of information, or too dense, which results in additional noise and complexity. Secondly, in case of dynamic background and moving camera, the detectors may extract irrelevant interest points that do not belong to an actual motion. To address these problems, a spatio-temporal interest point detector is designed to extract salient interest points within a region of interest where there is motion. In addition, a video stabilization is integrated in the detector to handle camera motion and dynamic background.

There are many approaches to represent local features e.g., traditional bag-of-words and super vector models. These approaches concatenate the features from multiple descriptors to get a large single vector for an entire video sequence. This concatenation does not retain spatio-temporal structure among the local feature descriptors. In addition, massive amount of data is generated using multiple feature descriptors from multiple modalities. This increases complexity and limits
many practical applications. To solve these problems, a tensor decomposition followed by a feature selection is applied. Tensor decomposition provides an efficient tool for discriminative feature extraction and model reduction by capturing multi-linear structures in high-order large-scale data.

In this work, firstly, we present a new method for visual feature extraction named refined dense trajectories. The refined dense trajectories method extracts salient interest points in a region of interest where there is motion, and discards the noisy and redundant interest points. The interest points are then tracked to form refined trajectories and visual features are computed along those trajectories. Secondly, we propose a novel spatio-temporal interest point detector based on a low-rank and group-sparse matrix approximation. The detector yields a set of salient spatio-temporal interest points which is neither too dense nor too sparse. To handle camera motion, a short-window video stabilization is integrated in the above visual feature extraction methods. The global motion is compensated by realigning of the video frames during interest point detection and trajectory formation. Thirdly, a unique super descriptor tensor decomposition model is presented for global feature representation. The local feature descriptors are first encoded through super descriptor vector coding and arranged in the form of tensors. Then discriminative features are obtained for classification through decomposition of rank-3 tensors followed by feature ranking. This approach retains the spatio-temporal structure among features from multiple descriptors and provides a significant dimensionality reduction.

The proposed visual feature extraction and bi-modal feature representation methods are evaluated through a detailed experimentation on multiple datasets: Maryland, YUPPEN, KTH, UCF, YouTube, TVHID, and Parliament. The proposed visual and audio-visual recognition systems are tested for the tasks of dynamic scene recognition, action recognition, human interaction recognition, and violent scene detection. The experimental results show that the proposed recognition systems outperform many state-of-the-art methods for visual and audio-visual recognition.
Acknowledgments

Firstly, I would like to thank my principal supervisor Professor Abdesselam Bouzerdoum and co-supervisor Associate Professor Son Lam Phung, for all their expert guidance, counsel, encouragement, technical, and moral support.

Secondly, I gratefully acknowledge the support provided by the staff of School of Electrical, Computer, and Telecommunications Engineering during my studies at University of Wollongong.

Thirdly, I would like to mention that this work has been supported in part by a grant from Australian Research Council.

Finally, I would like to express my gratitude to my parents, who have prayed and supported me throughout my studies. I thank to my fellow research students and friends, who have helped me during my studies at the university.
Chapter 1

Introduction

Chapter contents

1.1  Thesis Significance .................................................. 2
1.2  Research Objectives ............................................... 3
1.3  Thesis Contributions ................................................ 3
1.4  Thesis Organization ................................................ 4
1.5  Research Publications ............................................ 5

This research work addresses the problem of bi-modal audio-visual video understanding using computer vision and machine learning techniques. There are three major components in a general video recognition system: audio and visual feature extraction, global representation of the extracted features, and video classification. We focus on two important aspects in audio-visual video recognition: i) visual feature extraction from videos; and ii) global feature representation of local features for classification.

In this chapter, significance of the study, gaps in literature, and potential solutions are discussed in Section 1.1. The specific objectives of the thesis are given in Section 1.2. The contributions of this work are presented in Section 1.3. The thesis organization is given in Section 1.4. Finally, the publication outcomes of this research are listed in Section 1.5.
1.1 Thesis Significance

Video recognition depends highly on efficient visual feature extraction. Existing feature extraction methods are based mostly on spatio-temporal interest points detection in videos. The interest points are the key points where motion information is most discriminative. Local descriptors extract visual features within a volume, either around the interest points or along trajectories formed by tracking those interest points. The extracted local features are then encoded to obtain a global and meaningful representation for classification.

There are a few limitations associated with existing interest point detectors and global feature representation methods:

- The interest point detectors are either too sparse, which leads to loss of information, or too dense, which results in additional noise and complexity.

- In case of dynamic background and moving camera, the detectors may extract irrelevant interest points that do not belong to an actual motion.

- The exiting global feature representation methods simply concatenate the features from multiple feature descriptors and modalities, to get a large single vector for an entire video sequence. This destroys the spatio-temporal structure among the features and affects the classification accuracy.

- The concatenation of features from multiple descriptors and modalities yields a massive amount of data, which increases complexity and hinders many practical applications.

To address these problems, a spatio-temporal interest point detector is designed to extract salient interest points from regions where motion is the most dominant. Video stabilization is integrated into the detector to handle camera motion and dynamic background. Furthermore, a tensor decomposition followed by a feature selection is employed for discriminative feature extraction and dimensionality reduction by capturing multi-linear structures in high-order large-scale data. It is more efficient to arrange data from multiple descriptors in multi-dimensional arrays (i.e., tensors) instead of forming a large single vector.
1.2 Research Objectives

The specific aims of this research project are to:

1. Provide a review of audio-visual feature extraction, global feature representation, and video classification approaches for audio-visual video recognition problem.

2. Develop methods for visual feature extraction from videos to provide salient and discriminative features in presence of camera motion. Evaluate and compare performance of the developed visual feature extraction methods with existing methods for video recognition.

3. Develop a model for global representation of audio-visual features from multiple descriptors to preserve the spatio-temporal information among the features. Evaluate and compare performance of the developed global feature representation model with other methods for video recognition.

4. Test and evaluate the proposed recognition systems for applications of visual and audio-visual video recognition, and compare the performance with the state-of-the-art methods for the same tasks.

1.3 Thesis Contributions

The principal contributions of this thesis are listed as follows:

- A literature review on audio-visual recognition system is presented involving its individual components: audio-visual feature extraction, global feature representation, and video classification.

- A new method is proposed for visual feature extraction named refined dense trajectories. The refined dense trajectories method extracts salient interest points in a region of interest where there is motion and discards the noisy and redundant interest points.

- A novel spatio-temporal interest point detector based on a low-rank and group-sparse matrix approximation is presented. The detector yields a set
of salient spatio-temporal interest points, which is neither too dense nor too sparse.

- A short-window video stabilization is presented for the above visual feature extraction methods to handle camera motion. The global motion is compensated by realigning of the video frames during interest point detection and trajectory formation.

- A unique super descriptor tensor decomposition model is presented for global representation of bi-modal features. Discriminative features are obtained for classification through decomposition of tensor-based model followed by feature ranking. This retains the spatio-temporal structure among the features from multiple descriptors and modalities.

- The proposed video recognition system is applied to dynamic scene recognition, action recognition, violent scene detection, and human interaction recognition. The performance of the proposed recognition systems is compared with the state-of-the-art methods for the same tasks.

## 1.4 Thesis Organization

This thesis consists of seven chapters:

- **Chapter 1** outlines significance and objectives of the research project. It highlights the research contributions and publications.

- **Chapter 2** gives a literature review of the audio-visual video recognition and its components including audio-visual feature extraction, global feature representation, and video classification. This chapter discusses the different approaches available for the three components.

- **Chapter 3** presents the proposed methods for visual feature extraction, refined dense trajectories, low-rank and group-sparse matrix approximation based spatio-temporal interest point detector, and short-window video stabilization. Multiple dynamic scene and action recognition datasets are used to evaluate the proposed visual features extraction methods.
Chapter 4 presents the proposed super descriptor tensor decomposition model for global representation of audio-visual features. The chapter analyzes the individual components of the model for the tasks of dynamic scene recognition and human interaction recognition.

Chapter 5 details the classification results of the proposed recognition system for the applications of visual video recognition. This chapter compares the proposed recognition system with the state-of-the-art methods for dynamic scene recognition and action recognition.

Chapter 6 presents the classification results of proposed recognition system for the applications of audio-visual video recognition. The chapter evaluates and compares the proposed bi-modal recognition system with the state-of-the-art approaches for human interaction recognition and violent scene detection.

Chapter 7 summaries the research activities and provides the concluding remarks.

1.5 Research Publications

The publications arising from this research project are listed as follows:


Chapter 2

Review of Audio-visual Video Recognition

Chapter contents

2.1 Introduction ................................................. 8
2.2 Audio Feature Extraction ............................... 8
  2.2.1 Audio Attributes ................................. 9
  2.2.2 Audio Feature Descriptors .................... 10
2.3 Visual Feature Extraction ............................... 12
  2.3.1 Spatio-temporal Interest Point Detection .... 12
  2.3.2 Trajectory Formation .............................. 14
  2.3.3 Visual Feature Descriptors .................... 15
2.4 Global Feature Representation ....................... 17
  2.4.1 Vocabulary Generation ........................... 18
  2.4.2 Local Feature Encoding ........................... 19
  2.4.3 Pooling of Encoded Features ................... 21
2.5 Video Classification ................................. 22
  2.5.1 Instance based Learning .......................... 23
  2.5.2 Logic based Learning ............................ 24
  2.5.3 Statistical and Graphical Approaches .......... 24
  2.5.4 Support Vector Machines ....................... 25
  2.5.5 Neural Networks and Deep Learning .......... 25
2.6 Chapter Summary ........................................... 28
2.1 Introduction

Many recognition systems tend to exploit multi-modal information to achieve better performance [1]–[11]. Multiple modalities provide complementary information and one modality can give more useful information than the others. In addition, multiple modalities make a system more robust. Furthermore, unaffected modalities benefit a recognition system in presence of certain noise. For example, camera motion may affect the motion information but not the auditory cues which can be useful for the recognition task.

Audio-visual video recognition is an example of such a multi-modal recognition system, which uses more than one modality, and have appeared in different applications such as human interaction recognition [1], [2], action and event recognition [3], [4], and affect recognition [5]–[11]. In a general audio-visual video recognition system, firstly, local feature descriptors extract the audio and visual features from videos. Secondly, a global feature representation model encodes the extracted bi-modal features, and yields salient and discriminative features for classification. Thirdly, a classifier is used to perform the video recognition task.

This chapter provides an overview of different approaches for the three components in the above pipeline. Section 2.2 describes the different attributes of auditory information and some commonly used audio feature descriptors. Section 2.3 presents a review of approaches for local feature extraction, from spatio-temporal interest point detection, trajectory formation, to visual descriptor computation. Section 2.4 gives an overview of different global feature representation approaches, consist of vocabulary generation, local feature encoding, and pooling of encoded features. Section 2.5 provides a discussion about different classifiers that are commonly used for machine learning.

2.2 Audio Feature Extraction

Audio feature extraction deals with the extraction and analysis of audio signals to obtain a machine-processable representation. Audio features are as important as visual features for an efficient recognition system. For example, audio features extracted from the sounds of hand clapping, high five, and verbal communication
can be equally important as visual features [2]. Similarly, the sounds of gun-shots and explosions can be very informative for violence detection in automatic video surveillance [5].

There exists a huge amount of literature on audio feature extraction, see reviews in [12] and [13]. The audio features are usually developed for specific tasks, such as automatic speech recognition, sound recognition, audio segmentation, and music information retrieval. In this section, we describe a few audio attributes and descriptors for audio feature extraction.

2.2.1 Audio Attributes

The audio signals can be described in terms of different attributes such as duration, pitch, loudness, and timbre:

- **Duration** refers to start and end of an audio signal. Depending on sound envelope, the duration can be divided into four phases: attack, decay, sustain, and release. In some cases, silence can be of interest as well.

- **Pitch** mainly relates to frequency of a sound. We are usually interested in pitch strength which is defined as “subjective magnitude of the auditory sensation related to pitch” [14]. Pitch strength is determined by the spectral shape. Narrow-band noise and line spectra related sounds generate larger pitch strength than the broader spectral distribution signals.

- **Loudness** relates to the changes in sound pressure level. It is defined as the “attribute of auditory sensation in terms of which sounds can be ordered on a scale extending from soft to loud” [14].

- **Timbre** is the most complex attribute of sound. It is the “attribute of auditory sensation which enables a listener to judge that two non-identical sounds, similarly presented and having the same loudness and pitch, are dissimilar” [14]. In other words, timbre expresses the difference between hearing sensations of two different instruments, e.g., violin and piano, playing the same musical note.
Audio features represent the above-mentioned attributes. There is a wide range of audio feature descriptors that represent loudness and pitch. Other feature descriptors capture specific aspects of timbre such as tonality, frequency modulation, and sharpness. For a detailed overview, see [12] and [13].

### 2.2.2 Audio Feature Descriptors

We can categorize different audio feature descriptors based on their domains: temporal, frequency, and cepstral. A feature in frequency domain describes spectral characteristics of a signal, whereas a feature in temporal domain represents the signal’s waveform.

#### 2.2.2.1 Temporal Domain

The temporal domain describes the changes in the signal over time. Among many audio feature descriptors in temporal domain, zero crossing rate [15] and short-time energy [16] are commonly used. Zero crossing rate (ZCR) is one of the simplest feature descriptors, which gives the number of zero crossings within one second in temporal domain. Due to the simplicity of ZCR, it has been widely used for music classification, speech analysis, highlight detection, and environmental sound recognition. Short-time energy (SE) represents the envelope of a signal. SE can be defined as the per-frame mean energy which is also a measure for power. SE has mostly been used for audio retrieval.

#### 2.2.2.2 Frequency Domain

The frequency domain provides spectral distribution of a signal. It describes the harmonic structure, tonality, and bandwidth. There exist many descriptors in frequency domain [12], [13]. We consider here linear predictive coding [17] and spectral flux [18] descriptors due to their widespread use. Linear predictive coding (LPC) estimates the basic parameters of speech such as vocal tract transfer function and formant frequencies. LPC has been used for speech recognition, audio segmentation, and audio retrieval. The spectrum and cepstral representation of LPC are also used for recognition. Spectral flux (SF) is defined as the $\ell_2$-norm of difference vector of frame-to-frame spectral amplitude. SF quantifies the changes
2.2. Audio Feature Extraction

in spectrum shape over time. In SF, the flux is high for abrupt spectral changes like note onsets, and it is low for slowly changing spectral properties like noise. SF has been used in music and audio retrieval, music recognition, and speech analysis.

2.2.2.3 Cepstral Domain

The concept of cepstrum was introduced in [19]. Fourier transform of the logarithm of magnitude of the spectrum, provides representation in cepstral domain. Perceptual linear prediction [20] and Mel-frequency cepstral coefficients [21] are the commonly used audio feature descriptors in cepstral domain. Perceptual linear prediction (PLP) is based on the hearing concept, and it approximates the spectral shape using linear predictive analysis. PLP represents vocal tract characteristics and approximates many properties of human hearing. This gives PLP a better representation of spectral shape than LPC.

Mel-frequency cepstral coefficients (MFCCs) method has become one of the standard methods for audio retrieval and automatic speech recognition. To extract MFCCs, firstly, the audio signal is segmented into short overlapping frames. The reason for keeping the frames short is that the audio signal is assumed to be stationary over a short duration. The power spectrum of each frame is calculated using the periodogram. Then a Mel-filterbank with triangular filters is applied to the power spectra, and energy from each filter is obtained. To match the features closely to human hearing, the logarithms of all the filterbank energies are computed. As the filterbanks are usually overlapping in the frequency domain, a discrete cosine transform (DCT) is applied to the log-filterbank energies. In the end, a set of low frequency DCT coefficients is taken to represent the MFCCs. To exploit the discriminative ability of MFCCs, the first and second-order derivatives of MFCCs can also be used as features.

The different audio feature descriptors discussed above are listed in Table 2.1. The different audio feature descriptors are chosen depending on the type of information that needs to be extracted [12], [13]. Recently, MFCCs descriptor has been widely used for automatic speech recognition. The MFCCs collectively describe the coarse spectral shape, such as average power in spectrum, spectrum
2.3. Visual Feature Extraction

Table 2.1: Local descriptors for audio feature extraction.

<table>
<thead>
<tr>
<th>Feature Domain</th>
<th>Method</th>
<th>Comments</th>
<th>Author [ref]</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Low complexity, fast, and medium accuracy.</td>
<td>Zhang et al. [16]</td>
<td>2001</td>
</tr>
<tr>
<td>Frequency</td>
<td>LPC SF</td>
<td>Medium complexity, fast, and high accuracy.</td>
<td>Rabiner et al. [17]</td>
<td>1978</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Low complexity, fast, and medium accuracy.</td>
<td>Scheirer et al. [18]</td>
<td>1997</td>
</tr>
<tr>
<td>Cepstral</td>
<td>PLP</td>
<td>High complexity, medium speed, and medium accuracy.</td>
<td>Hermansky et al. [20]</td>
<td>1990</td>
</tr>
<tr>
<td></td>
<td>MFCCs</td>
<td>High complexity, medium speed, and high accuracy.</td>
<td>Davis et al. [21]</td>
<td>1980</td>
</tr>
</tbody>
</table>

centroid, pitch, and tone. Due to a diverse representation of audio signals, MFCCs descriptor is considered one of the state-of-the-art methods.

2.3 Visual Feature Extraction

In a video recognition system, visual features (static or dynamic) play the most important role for efficient recognition. The visual feature extraction mostly starts from the detection of spatio-temporal interest points (STIPs) in videos. Then, visual descriptors are computed within a volume, either around the STIPs [22] or along the trajectories formed by tracking those STIPs [23]. In this section, we discuss different methods for STIP detection, trajectory formation, and local descriptors for visual feature extraction in videos.

2.3.1 Spatio-temporal Interest Point Detection

The spatio-temporal interest points are the key points in the space-time, where the visual features are most discriminative. There exist many approaches for STIP detection [22], [24]–[31], which can be categorized as spatio-temporal corner detectors [22], [24]–[25], spatio-temporal filtering methods [26]–[28], and global information based techniques [29]–[31].

2.3.1.1 Spatio-temporal Corner Detectors

The spatio-temporal corner detectors are extensions of 2D corner detectors in the time domain. In [22], Harris3D detector was presented which is an extension of 2D Harris corner detector to the space-time domain. In Harris3D, regions with high intensity variations are detected as a sparse set of STIPs. In [24], Hes-STIP detector was proposed which is a spatio-temporal extension of 2D scale-invariant
Harris-Laplace corner detector. In Hes-STIP, a dense set of STIPs is extracted using Hessian saliency measure. In [25], V-FAST corner detector was presented to detect STIPs by extending FAST corner detector to the time domain. Spatio-temporal saliency is detected when several contiguous pixels on a circle are brighter than a reference pixel, yielding in desired STIPs. The corner detection based methods usually result in a sparse set of STIPs, which may lead to loss of information.

2.3.1.2 Spatio-temporal Filtering Methods

The spatio-temporal filtering methods employ Gaussian and Gabor filters in the space-time domain to detect STIPs. In [26], a 2D Gaussian filter in the space and a 1D Gabor filter in the time domain are used to detect STIPs. This approach was extended in [27] using 2D Gabor filters in the space-time domain. Like corner detectors, these approaches also yield a sparse set of STIPs. They focus on local spatio-temporal information instead of global motion, which results in unwanted STIPs caused by camera motion. In another approach [28], a selective set of STIPs is detected by applying temporal constraints based on 2D Gabor filters, and a STIP matching algorithm is used to remove camera motion.

2.3.1.3 Global Information based Techniques

The global information is incorporated to detect STIPs in [29]–[31]. In [29], a non-negative matrix factorization (NNMF) based detector was proposed, which uses global information of moving points. The STIPs are detected by considering their relation to the relevant motion. In [30], a 3D transform was presented for capturing global distribution of STIPs using Radon features. Similarly, in [31], histogram of interest points (HIPL) method was introduced for capturing information about the spatial distribution of STIPs.

The choice of a detector to extract STIPs depends on the nature of videos. In some cases, extracting a sparse set of STIPs helps, whereas in other cases, a dense set of STIPs may capture more information. For the videos with natural scenes, going towards one extreme may not be useful. For example, a detector being too sparse can lead to loss of information, whereas a detector being too dense may add noise and complexity [32]. Table 2.2 lists the various spatio-temporal interest
Table 2.2: Spatio-temporal interest point detectors for videos.

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Comments</th>
<th>Author [ref]</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corner detection</td>
<td>Harris3D</td>
<td>Sparse, low detection rate, and fast.</td>
<td>Laptev et al. [22]</td>
<td>2005</td>
</tr>
<tr>
<td></td>
<td>Hes-STIP</td>
<td>Dense and medium detection rate and speed.</td>
<td>Willems et al. [24]</td>
<td>2008</td>
</tr>
<tr>
<td></td>
<td>V-FAST</td>
<td>Sparse, low detection rate, and fast.</td>
<td>Yu et al. [25]</td>
<td>2010</td>
</tr>
<tr>
<td>Space-time</td>
<td>Cuboids</td>
<td>Sparse, low detection rate, and fast.</td>
<td>Dollar et al. [26]</td>
<td>2005</td>
</tr>
<tr>
<td>filtering</td>
<td>E-cuboids</td>
<td>Sparse and medium detection rate and speed.</td>
<td>Bregonzio et al. [27]</td>
<td>2008</td>
</tr>
<tr>
<td></td>
<td>Selective STIPs</td>
<td>Dense, high detection rate, and slow.</td>
<td>Chakraborty et al. [28]</td>
<td>2012</td>
</tr>
<tr>
<td>Global</td>
<td>NNMF</td>
<td>Sparse, medium detection rate, and fast.</td>
<td>Wong et al. [29]</td>
<td>2007</td>
</tr>
<tr>
<td>information</td>
<td>3D R-Transform</td>
<td>Sparse, low detection rate, and slow.</td>
<td>Yuan et al. [30]</td>
<td>2013</td>
</tr>
<tr>
<td></td>
<td>HIPL</td>
<td>Sparse, low detection rate, and fast.</td>
<td>Yan et al. [31]</td>
<td>2012</td>
</tr>
</tbody>
</table>

2.3.2 Trajectory Formation

The spatio-temporal interest points provide the key points for feature extraction. It is easy to simply take a volume around the STIPs, and then calculate the visual features descriptors within that volume [22]. This may not provide all the necessary space-time information that is needed. It is better to form motion trajectories by tracking the STIPs in consecutive frames of a video. The local descriptors can then be calculated within a volume along those trajectories [23].

We consider here a few interest point tracking methods to form motion trajectories. In [33], a Kanade-Lucas-Tomasi (KLT) algorithm was presented to locate and track the interest points. In KLT algorithm, the minimum eigen-values of gradient matrices are used to detect the interest points. The interest points are then tracked using the Newton-Raphson method. The use of KLT algorithm was also seen in [34] and [35]. In [36], the interest points are detected using FAST corner detector, and then tracked by matching histogram of oriented gradient (HOG) descriptors over the consecutive frames. The trajectories were shown to be less sensitive to noise in comparison with the trajectories formed by KLT algorithm. In [37], the interest points are tracked by pair-wise matching of scale-invariant feature transform (SIFT) descriptors over consecutive frames. It was shown that the SIFT tracking based trajectories achieved better classification accuracy than KLT trajectories. Later in [38], the trajectories from KLT and SIFT tracking are combined to formulate visual matching and tracking. In [23], dense trajectories
2.3. Visual Feature Extraction

Table 2.3: Various approaches for trajectory formation in videos.

<table>
<thead>
<tr>
<th>Method</th>
<th>Tracking based on</th>
<th>Comments</th>
<th>Author [ref]</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>KLT tracker</td>
<td>Newton-Raphson method</td>
<td>Medium quality trajectories, some irregular patterns, and medium complexity.</td>
<td>Lucas et al. [33]</td>
<td>1981</td>
</tr>
<tr>
<td>HOG tracker</td>
<td>HOG descriptor matching</td>
<td>Low quality trajectories, irregular patterns, and medium complexity.</td>
<td>Kaaniche et al. [36]</td>
<td>2009</td>
</tr>
<tr>
<td>SIFT tracker</td>
<td>SIFT descriptor matching</td>
<td>Medium quality trajectories, some irregular patterns, and medium complexity.</td>
<td>Sun et al. [37]</td>
<td>2009</td>
</tr>
<tr>
<td>KLT-SIFT tracker</td>
<td>KLT+SIFT tracking</td>
<td>Medium quality trajectories, some irregular patterns, and high complexity.</td>
<td>Sun et al. [38]</td>
<td>2010</td>
</tr>
<tr>
<td>Dense Trajectories</td>
<td>Dense optical flow field</td>
<td>High quality trajectories, smooth patterns, and high complexity.</td>
<td>Wang et al. [23]</td>
<td>2013</td>
</tr>
</tbody>
</table>

method was proposed to form trajectories by tracking densely sampled interest points. In dense trajectories method, the interest points are tracked using dense optical flow field. It has been shown that dense trajectories outperform KLT and SIFT matching based trajectories [23]. The various trajectory formation approaches are listed in Table 2.3.

2.3.3 Visual Feature Descriptors

The visual feature descriptors are used to extract appearance and motion information from videos. Here we discuss some existing local feature descriptors proposed for videos. The visual feature descriptors can be categorized based on appearance information [24], [39]–[41], and motion information [23], [26], [39], [43], [44]. These approaches are discussed as follows.

2.3.3.1 Appearance based Visual Descriptors

The appearance information around interest points or trajectories can be very discriminative for video representation. There exist different visual feature descriptors which make use of gradient orientations to extract appearance information. In [39], histogram of oriented gradients descriptor was proposed for videos, which is a variant of HOG descriptor for images initially proposed in [45]. For videos, the HOG descriptor captures the shape and appearance information, either around an interest point or along a trajectory within a spatio-temporal grid. The edge orientations are computed and quantized into histogram bins. A histogram is calculated in each cell of the spatio-temporal grid, which is then normalized and concate-
nated to obtain visual features. In [40], 3-dimensional HOG (HOG3D) descriptor was presented by extending HOG image descriptor [45] to the spatio-temporal domain. Based on convex regular polyhedrons, HOG3D computes 3D gradient orientations, which are then quantized to form histograms. A similar approach to HOG3D is called histogram of oriented 4-dimensional normals (HON4D), which combines 3D surface normals with time [41]. The 3D depth maps are used as the basis of descriptor computation, rather than 2D image frames as in HOG3D.

There exist some visual feature descriptors which also extend 2D image descriptors to the space-time domain for videos. For example, in [42], 3-dimensional scale-invariant Fourier transform (3DSIFT) descriptor was presented, which is an extension of SIFT descriptor [46] for images to the spatio-temporal domain. Based on the concept of spatio-temporal grids and gradients, 3DSIFT weights each pixel by a Gaussian. The Gaussian weighting gives less importance to those gradients which are far away from the center of local features. A dominant orientation is determined which is used to make the descriptor rotation-invariant. In [24], extended speeded-up robust features (E-SURF) descriptor was proposed, which is an extension of SURF descriptor [47] to the spatio-temporal domain. The space-time volume that surrounds an interest point is divided into a spatio-temporal grid. Haar-wavelets are used to obtain the local features by representing each cell in the grid by a vector.

### 2.3.3.2 Motion based Visual Descriptors

The early works on visual feature descriptors in videos were presented in [43] and [26]. In [43], multiple feature descriptors were proposed by representing motion as spatio-temporal jets, position independent and dependent histograms, and principal component analysis (PCA), calculated for optical flow and spatio-temporal gradients. In [26], several descriptors were presented based on transformations of local neighborhoods such as windowed optical flow, normalized pixel values, and brightness gradient. The features are obtained by taking histogram of the values and flattening of the local neighborhood in small grids. In this approach, PCA is used for dimensionality reduction. The shape of motion trajectories also leads to visual feature extraction. For example, in [23], a trajectory shape descriptor was
proposed to capture and encode the shape of trajectories. To describe the shape of a trajectory, a sequence of displacement vectors is normalized using the sum of displacement vector magnitudes.

The commonly used visual descriptors based on optical flow are histogram of optical flow (HOF) \[39\] and motion boundary histograms (MBH) \[44\]. HOF was proposed to encode motion information in videos. HOF first calculates the optical flow, then quantizes the flow information into histogram bins to obtain visual features. Rather than using simple optical flow, MBH descriptor computes derivatives of horizontal and vertical components of the optical flow to encode the relative motion between pixels. MBH descriptor is more robust to camera motion than the normal optical flow. This is because MBH represents the gradient of optical flow. In \[23\], MBH was employed to extract local features along motion trajectories. In this approach, spatial derivatives are computed for horizontal and vertical components of the optical flow resulting in MBHx and MBHy descriptors. A histogram of each component is obtained and then normalized using $\ell_2$-norm to obtain the visual features.

The widely used visual feature descriptors are HOG and HOF, which capture appearance and motion information, respectively. The HOG descriptor computes the orientation of shape at the finest level (e.g., each pixel) to capture the appearance information. Although HOF has been used extensively, it is sensitive to camera motion. Recently, MBH has been used widely because it is based on derivatives of optical flow, which helps with suppressing the camera motion \[23\]. We list various appearance and motion based visual feature descriptors in Table 2.4.

### 2.4 Global Feature Representation

The local audio and visual features are usually processed to obtain a global representation for classification. The most popular representation is bag-of-words (BoW) originally proposed to represent text for document retrieval \[48\]. Since then the BoW model has been studied widely for information retrieval, natural language processing, and computer vision. The BoW model encodes the
Table 2.4: Visual feature descriptors for videos.

<table>
<thead>
<tr>
<th>Method</th>
<th>Descriptor based on</th>
<th>Comments</th>
<th>Author [ref]</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOG3D</td>
<td>3D gradients</td>
<td>Medium classification accuracy and medium complexity.</td>
<td>Klaser et al. [40]</td>
<td>2008</td>
</tr>
<tr>
<td>HON4D</td>
<td>4D orientation normals</td>
<td>Medium classification accuracy and high complexity.</td>
<td>Oreifej et al. [41]</td>
<td>2013</td>
</tr>
<tr>
<td>3DSIFT</td>
<td>Spatio-temporal gradients</td>
<td>Medium classification accuracy and medium complexity.</td>
<td>Scovanner et al. [42]</td>
<td>2007</td>
</tr>
<tr>
<td>E-SURF</td>
<td>Haar wavelets</td>
<td>Low classification accuracy and high complexity.</td>
<td>Willems et al. [24]</td>
<td>2008</td>
</tr>
<tr>
<td>Space-time jets</td>
<td>Gradients and optical flow</td>
<td>High classification accuracy and medium complexity.</td>
<td>Laptev et al. [43]</td>
<td>2006</td>
</tr>
<tr>
<td>Cuboids</td>
<td>Gradients and optical flow</td>
<td>Low classification accuracy and medium complexity.</td>
<td>Dollar et al. [26]</td>
<td>2005</td>
</tr>
<tr>
<td>Trajectory-shape</td>
<td>Shape of trajectories</td>
<td>Medium classification accuracy and low complexity.</td>
<td>Wang et al. [23]</td>
<td>2013</td>
</tr>
</tbody>
</table>

global statistics of local features by calculating histogram of feature occurrences in videos. The BoW model consists of three major components: i) vocabulary generation, ii) local feature encoding, and iii) pooling and normalization of encoded features. The vocabulary is created through unsupervised learning of local features from training video sequences. The feature encoding generally represents the local features using some coding method to obtain the codewords. The final features for classification are then obtained by pooling and normalization of codewords. In this section, we discuss various approaches for the three components of the BoW model.

### 2.4.1 Vocabulary Generation

A vocabulary divides the feature space into several regions or clusters. The vocabulary sometimes is also called codebook or dictionary. Local features in a region relate to a codeword represented by an integer (ranging from 1 to vocabulary size). The local features are later encoded as a histogram of codewords.

There exist many approaches to compute the vocabulary. For example, in [49], k-means algorithm was proposed to compute the vocabulary. A codeword is considered as the cluster center in k-means. The cluster center is the mean of all feature vectors which belong to that codeword. In k-means, the clusters are positioned exclusively around the densest regions in feature space. This does not code other informative regions. To overcome this drawback, in [50], a fixed radius clusterer method based on mean shift (MS) was proposed to generate the vocabulary. It was shown that mean shift based clustering performed better than the k-means clustering. In [51], an information theoretic method based on min-
Table 2.5: Vocabulary generation methods for global feature representation.

<table>
<thead>
<tr>
<th>Method</th>
<th>Comments</th>
<th>Author [ref]</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-means</td>
<td>Average clustering and low complexity.</td>
<td>Sivic et al. [49]</td>
<td>2003</td>
</tr>
<tr>
<td>MS</td>
<td>Good clustering and medium complexity.</td>
<td>Jurie et al. [50]</td>
<td>2005</td>
</tr>
<tr>
<td>MIL</td>
<td>Good clustering and medium complexity.</td>
<td>Lazebnik et al. [51]</td>
<td>2009</td>
</tr>
<tr>
<td>RL</td>
<td>Average clustering and medium complexity.</td>
<td>Tuytelaars et al. [52]</td>
<td>2007</td>
</tr>
<tr>
<td>SC</td>
<td>Excellent clustering and high complexity.</td>
<td>Yang et al. [53]</td>
<td>2009</td>
</tr>
<tr>
<td>GMM</td>
<td>Excellent clustering and high complexity.</td>
<td>Winn et al. [54]</td>
<td>2005</td>
</tr>
</tbody>
</table>

Imization of information loss (MIL) was proposed to simultaneously learn the vocabularies in the Euclidean feature space. This approach captures the components that are semantically common. In [52], a data-independent approach was presented that divides the feature space into a regular lattice (RL) for construction of the vocabulary. The hashing techniques are used to store only non-empty bins, and fit the method to fine-grained grids which accommodates the high dimensional feature space. This is different from learning the division of feature space from the training data.

There are some approaches which tend to outperform the above-mentioned approaches for vocabulary generation. For example, in [53], sparse coding (SC) is used to generate the vocabulary. It was shown that sparse coding performed better than $k$-means based vocabulary learning. In [54], Gaussian mixture model (GMM) is used to represent the local features. The centers of the Gaussian components represent the codewords. Although GMM tend to have more representation power in comparison with a single cluster center, it needs higher computational resources. The various methods for vocabulary generation are listed in Table 2.5.

### 2.4.2 Local Feature Encoding

The local audio and visual features are encoded to obtain a more meaningful representation. The local features can be represented either as a combination of codewords obtained after feature encoding [53], [55]–[58], or as differences between local features and the codewords [59]–[61]. A detailed description and comparison of different feature encoding techniques are provided in [62] and [63].

There exist many feature encoding techniques, here we discuss a few methods
that represent the local features as a combination of codewords. In [55], a vector quantization (VQ) method was proposed to encode the local features from images. In this method, the vocabulary is generated using $k$-means algorithm. The local features are then quantized through hard-assignment of the features to the vocabulary. Hard-assignment can be restrictive in representing the features. Many approaches replace the hard-assignment with alternative feature encoding to retain more information about the features. For example, in [56], soft-assignment of local features to the vocabulary was presented. A kernel codebook encoding (KCB) was proposed which associates the local features with multiple nearby codewords, rather than a single nearest codeword. The local features are mapped to weighted combination of codewords. In another work [57], locality-constrained linear coding (LLC) was proposed to encode the local features by projecting the features to the local linear subspace. The subspace consists of multiple closest codewords. The feature representation is obtained by max pooling of the coordinates resulted from the projection of each local feature into its local coordinate system. It was shown that LLC performed better in comparison with VQ and KCB methods.

The above-mentioned feature encoding techniques extract order-less features. In [58], spatial pyramid matching (SPM) was proposed to retain the global geometric correspondence of images. In SPM, the images are divided into regular grids and the local features are computed in each grid. The vocabulary is generated using $k$-means algorithm and the features are encoded using VQ encoding. The final features are obtained through average pooling of the encoded features. In [53], a similar approach based on sparse coding was presented which is a variation of SPM. In this approach, $k$-means vector quantization is replaced by the sparse coding for quantization of local features, and the average pooling is replaced by the max pooling. The sparse coding based SPM (ScSPM) outperformed many previous feature encoding techniques including VQ, KCB, and SPM.

Recently, different encoding techniques have been developed which represent the local features as differences between the features and the codewords. In [59], a super-vector coding (SVC) method was proposed to encode the local features. SVC uses hard-assignment of the local features to single nearest codeword. SVC
2.4. Global Feature Representation

Table 2.6: Representative feature encoding techniques based on combination of codewords.

<table>
<thead>
<tr>
<th>Method</th>
<th>Comments</th>
<th>Author [ref]</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>VQ</td>
<td>Low accuracy and low complexity.</td>
<td>Csurka et al. [55]</td>
<td>2004</td>
</tr>
<tr>
<td>KCB</td>
<td>Low accuracy and low complexity.</td>
<td>Gemert et al. [56]</td>
<td>2008</td>
</tr>
<tr>
<td>LLC</td>
<td>Medium accuracy and high complexity.</td>
<td>Wang et al. [57]</td>
<td>2010</td>
</tr>
<tr>
<td>SPM</td>
<td>Medium accuracy and medium complexity.</td>
<td>Lazebnik et al. [58]</td>
<td>2006</td>
</tr>
<tr>
<td>ScSPM</td>
<td>High accuracy and high complexity.</td>
<td>Yang et al. [53]</td>
<td>2009</td>
</tr>
</tbody>
</table>

also uses soft-assignment of the local features to several nearest codewords. In SVC, the local features are represented as first-order differences between the features and the codewords. In [60], vector of locally aggregated descriptors (VLAD) was proposed for feature encoding. The encoded features are obtained by matching each local feature vector to its closest codeword. The final features are obtained by averaging of differences between descriptors assigned to the clusters and their centroids. In [61], Fisher vector (FV) encoding was proposed to represent the local features same as SVC (i.e., using differences between features and code words). In FV, the vocabulary is generated using GMM instead of \( k \)-means. The local features are represented by capturing first and second-order differences between the features and codewords (Gaussian components). There also exist some variants of FV encoding called improved FV (IFV) and stacked FV (SFV), in [64] and [65], respectively.

The feature encoding techniques like VQ, LLC, and SPM are commonly used for feature encoding because their representation requires a small storage capacity. Techniques like SVC, FV, and VLAD have powerful representation but they require a large storage capacity. So far, FV encoding is the state-of-the-art for local feature encoding [62], [63]. The various feature encoding techniques for global feature representation are listed in Tables 2.6 and 2.7.

2.4.3 Pooling of Encoded Features

The local features after the feature encoding are pooled towards obtaining a compact final representation. There are three common feature pooling techniques: sum pooling, average pooling, and max pooling. In sum pooling, the encoded
Table 2.7: Representative feature encoding techniques based on difference between features and codewords.

<table>
<thead>
<tr>
<th>Method</th>
<th>Comments</th>
<th>Author [ref]</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC</td>
<td>Medium accuracy and medium complexity.</td>
<td>Zhou et al. [59]</td>
<td>2009</td>
</tr>
<tr>
<td>VLAD</td>
<td>Medium accuracy and high complexity.</td>
<td>Jegou et al. [60]</td>
<td>2010</td>
</tr>
<tr>
<td>FV</td>
<td>High accuracy and high complexity.</td>
<td>Sanchez et al. [61]</td>
<td>2013</td>
</tr>
<tr>
<td>SFV</td>
<td>High accuracy and high complexity.</td>
<td>Peng et al. [65]</td>
<td>2014</td>
</tr>
</tbody>
</table>

feature vectors are added to obtain a single vector which represents a video sequence, \( p_j = \sum_{i=1}^{N} e_{i,j} \), where \( e_i \in \mathbb{R}^M \), \( i = 1, ..., N \), represents the encoded vector, and \( p_j, j = 1, ..., M \), represents the \( j \)th entry in the pooled vector. This pooling strategy is intuitive and has been used on multiple occasions [66], [67]. In average pooling, the resultant vector from the sum pooling is further divided by the total number of encoded vectors, \( p_j = \sum_{i=1}^{N} e_{i,j}/N \). Although average pooling has been used in some methods [68], it is not considered the best pooling technique [69]. In max pooling, the maximum value (element wise) is picked from each encoded vector to form a single vector, \( p_j = \max e_{i,j}, i \in \{1,...,N\} \). Max pooling is a widely accepted technique and used in many methods [70], [71]. Once the encoded feature vectors are pooled, the resultant vector can be normalized using \( \ell_1 \), \( \ell_2 \), and power normalization. Sometimes two normalizations are combined like \( \ell_2 \) and power normalization [66].

### 2.5 Video Classification

The video classification is the final step in a recognition framework. The features obtained after global feature representation are fed to a machine learning algorithm, which classifies the features into different categories or classes. In general, the existing machine learning algorithms can be categorized as supervised, unsupervised, and semi-supervised learning methods. This categorization is based on label information that comes with the features. In supervised learning, the input training features carry the label information. A model is trained to make predictions, and the model is corrected if it makes a wrong prediction. To achieve a desired level of accuracy, the training process is continued on the
training data. The supervised learning is used for classification and regression problems. In *unsupervised learning*, the training features do not include the label information. A model examines the data structure to extract general rules. These rules are then used to organize the features according to their similarities. Unsupervised learning is generally used for clustering (vocabulary generation), association rule learning, and dimensionality reduction. In *semi-supervised learning*, the input training features have mixed labeled and unlabeled information. A model must perform both the tasks: feature organization and label prediction. Semi-supervised learning is used for classification and regression problems.

The training features usually come with the label information for a video recognition problem. Therefore, we focus only on supervised learning for video classification. The goal is to build a model that can capture the distribution of class labels in terms of training features. Then the trained model is used for prediction and assignment of labels to test features. There are many supervised learning algorithms for classification problem, we therefore discuss commonly used classification algorithms. The various algorithms can be categorized as instance based learning algorithms [72]–[74], logic based learning algorithms [75]–[78], statistical and graphical approaches [79]–[85], support vector machines [86]–[90], and neural network and deep learning methods [91]–[97].

### 2.5.1 Instance based Learning

The most popular instance based learning classification method is *k*-nearest neighbor (*k*NN) [72]. For a given test instance, the *k*NN first locates the *k* nearest instances, then a label is determined based on single most frequent label of the nearest instance. The *k*NN classifier needs to store all the instances. Also, it is sensitive to irrelevant features and the similarity functions chosen for comparison of the instances. There exist some variants of *k*NN classifier. For example, in [73], a condense nearest neighbor algorithm was proposed for dataset reduction. A set of prototypes is selected from the training data for classification for same accuracy as *k*NN with whole dataset. In [74], a universal nearest neighbor algorithm was presented which induces a leveraged *k*NN rule. This rule weights the votes of nearest neighbors through leveraging coefficients. The coefficients are learned
iteratively from the training data.

### 2.5.2 Logic based Learning

Logic based learning leads to decision tree classifiers which classify features based on sorting of feature values. In decision trees, a node represents a test on a feature, a branch yields an outcome for the test, and a leaf node gives a class label. To construct a decision tree, best features are found through different measures such as gain ratio, information gain, Gini index, and Chi-square. These measures may not be significantly better than each other. The making of an optimal decision tree is a non-deterministic polynomial-time (NP) complete problem. The widely used decision trees include iterative dichotomiser-3 (ID3) [75], classification and regression trees (CART) [76], and C4.5 [77].

A limitation of decision trees is overfitting of the training data. To handle this limitation, random forests were proposed in [78]. Based on a collection of several individual trees, random forest predicts a class label through voting on the results from all the individual trees. In random forest, first, bootstrapping of original sample data is performed to produce the training set for individual trees. Then for each decision tree, a bagging process is done after training over the bootstrapped data. Random forest selects a few features to grow by expanding at each node. Random forest is flexible and has a few parameters to be tuned.

### 2.5.3 Statistical and Graphical Approaches

There are different classifiers which make use of statistical measures like probability and conditional probability. For example, naive Bayes (NB) classifier provides a probability that a given data sample belongs to a specific class [79]. NB classifier is based on Bayes’ rule. It makes a naive (strong) assumption that all variables are statistically independent and contribute towards classification. Another classifier is Bayesian network, which uses a directed acyclic graph to represent a set of random variables and their conditional dependencies. A Bayesian network is a probabilistic graphical model where each feature corresponds to one node. A Bayesian network unrolled in the time axis is dynamic Bayesian network (DBN) [80]. DBN contains multiple random variables in its state space.
There exist some other graphical models which have been widely used for classification. For example, a simplified version of DBN with fixed graph structure and only one random variable is hidden Markov model (HMM) [81]. HMM is a generative model which assumes that the model is a Markov process with hidden (unobserved) states. HMM models the transition matrix based on the training data to give the output. HMM assumes that the observations are independent given their labels. This assumption is violated if the observations have complex features. Whereas, this assumption is abandoned in [82] by conditioning on the entire observations. In [82], conditional random field (CRF) was presented, which is a discriminative model and a generalization of HMM [82]. CRF is an undirected graphical model which gives the conditional probability of a label sequence given a sequence of observations. CRF has more discriminative power than HMM, and it outperforms HMM for classification purpose [82]. There also exist some variants of CRF for the recognition task in computer vision, e.g., hidden CRF [83], multi-scale CRF (mCRF) [84], and latent dynamic CRF (LDCRF) [85].

2.5.4 Support Vector Machines

Support vector machines (SVM) classifier is widely used for classification problem [86]–[90]. SVM constructs a hyperplane in high-dimensional space, which separates features of two classes on either side of it. The hyperplane achieves a good separation if it maximizes its distance to the nearest data point of any class. The margin should be large to reduce the generalization error. SVM classifier is generally fast and yields good classification results. The original SVM was designed in [86], which is a linear classifier. Later in [89], different kernel tricks were applied which resulted in a non-linear SVM classifier. Some widely used kernel functions include polynomial, sigmoid, and radial basis function. There also exist some extensions of SVM such as multi-class SVM, transductive SVM, structured SVM, and Bayesian SVM.

2.5.5 Neural Networks and Deep Learning

Artificial neural network (ANN) has achieved a great attention for the task of classification and recognition [91]–[97]. ANN consists of a group of connected
units called neurons organized into multiple layers: input layer, hidden layers, output layer. Input layer is made of input neurons, and it receives the information that needs to be processed. Hidden layers are made of hidden neurons, and they process the data. Output layer is made of output neurons, and it yields the results of the network. To map the input to the output, ANN learns the weights on the connections between the neurons. The learning of weights is usually time consuming, the learning time is increased even further for multiple hidden layers. For a better performance, the hidden layers have parameters which usually need to be tuned.

There exist different types of ANN such as extreme learning machines (ELM) [92], deep neural network (DNN) [93], and convolutional neural network (CNN) [94]. Extreme learning machines were originally presented for generalized single-hidden layer feedforward networks. An important aspect of ELM is that it has only one hidden layer of neurons which needs not to be tuned. This is different from general network structure of neural networks where parameter tuning is required for hidden layers. The learning using ELM is much faster and the training error is much smaller than common neural networks. Since the hidden layer in ELM needs not to be tuned, and its parameters can be fixed, the output weights can be resolved via least-square method. In comparison, a deep neural network has multiple hidden layers of neurons between the input and the output layers. DNN can have a large number of hidden layers where every neuron in one layer is connected to a neuron in the next layer. This leads to overfitting and very slow learning. DNN can be applied on raw input data, and it learns features directly from the data rather than extracting features manually. CNN is similar to DNN but its hidden layers contain special layers called convolutional and pooling layers. These layers apply convolution and pooling operations on patches of neurons in one layer and pass the output to the next layer. Since only a patch of neurons in one layer is connected to a neuron in next layer, the number of connections between hidden layers is less than that of in DNN. CNN has recently been used for image and video classification. For example, fast feature embedding using convolutional networks for images [95], learning of spatio-temporal features using 3D convolutional networks for videos [96], and
### 2.5. Video Classification

Table 2.8: Machine learning methods for video classification.

<table>
<thead>
<tr>
<th>Category</th>
<th>Method</th>
<th>Comments</th>
<th>Author [ref]</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance based learning methods</td>
<td>k-NN</td>
<td>Low accuracy, low complexity, and fast.</td>
<td>Cover et al. [72]</td>
<td>1967</td>
</tr>
<tr>
<td></td>
<td>Condensed-NN</td>
<td>Low accuracy, low complexity, and fast.</td>
<td>Hart [73]</td>
<td>1968</td>
</tr>
<tr>
<td></td>
<td>Universal-NN</td>
<td>Medium accuracy, medium complexity, and fast.</td>
<td>Nock et al. [74]</td>
<td>2012</td>
</tr>
<tr>
<td>Logic based learning methods</td>
<td>ID3</td>
<td>Low accuracy, low complexity, and fast.</td>
<td>Quinlan [75]</td>
<td>1986</td>
</tr>
<tr>
<td></td>
<td>CART</td>
<td>Medium accuracy, complexity, and speed.</td>
<td>Breiman et al. [76]</td>
<td>1984</td>
</tr>
<tr>
<td></td>
<td>C4.5</td>
<td>Medium accuracy, complexity, and speed.</td>
<td>Quinlan [77]</td>
<td>1993</td>
</tr>
<tr>
<td></td>
<td>Random forest</td>
<td>Medium accuracy, high complexity, and slow.</td>
<td>Breiman [78]</td>
<td>2001</td>
</tr>
<tr>
<td></td>
<td>DBN</td>
<td>Low accuracy and medium complexity and speed.</td>
<td>Luo et al. [80]</td>
<td>2003</td>
</tr>
<tr>
<td></td>
<td>HMM</td>
<td>Medium accuracy, high complexity, and slow.</td>
<td>Rabiner [81]</td>
<td>1989</td>
</tr>
<tr>
<td></td>
<td>CRF</td>
<td>Medium accuracy, high complexity, and slow.</td>
<td>Lafferty et al. [82]</td>
<td>2001</td>
</tr>
<tr>
<td>Support vector machines</td>
<td>Linear SVM</td>
<td>High accuracy, medium complexity, and fast.</td>
<td>Vapnik [86]</td>
<td>1963</td>
</tr>
<tr>
<td></td>
<td>Non-linear SVM</td>
<td>High accuracy, high complexity, and medium speed.</td>
<td>Boser et al. [89]</td>
<td>1992</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Vapnik [90]</td>
<td>1998</td>
</tr>
<tr>
<td>Neural networks and deep learning</td>
<td>ELM</td>
<td>High accuracy and medium complexity and speed.</td>
<td>Huang et al. [92]</td>
<td>2004</td>
</tr>
<tr>
<td>methods</td>
<td>DNN</td>
<td>High accuracy, high complexity, and slow.</td>
<td>Deng et al. [93]</td>
<td>2014</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>High accuracy, high complexity, and slow.</td>
<td>LeCun et al. [94]</td>
<td>1995</td>
</tr>
</tbody>
</table>

The various machine learning methods for video classification are listed in Table 2.8. The performance of different classifiers depends on the structure and information hidden in the input features. $k$NN is a widely used classifier for performance comparison of different recognition methods because it does not involve special parameters to be tuned. NB is useful when the training data is categorical. NB usually needs a large amount of training data for better performance. Decision trees are robust to missing values and errors in the training data. Decision trees are easy to use but they lack accuracy when the trees get large. SVM does not overfit the training data and is considered a robust and accurate method. However, the performance of SVM is highly dependent on the selection of an appropriate kernel function. Generally, SVM classifier performs better than $k$NN, naive Bayes, and decision tree classifiers [98]. Graphical methods usually perform well to capture the structure in data, but they are complex and take more computation time. Neural networks and deep learning methods have achieved the state-of-the-art performance for classification task. For example, ELM outperforms the SVM but it takes more computation resources and time if the number of neurons is increased [99]. CNN tends to outperform the other methods for classification but its learning process is time consuming.
2.6 Chapter Summary

In this chapter, we provided a detailed literature review on audio-visual recognition systems and its components: local feature extraction (audio and visual), global feature representation, and video classification. Firstly, we discussed different audio attributes and local descriptors for audio feature extraction. Secondly, we reviewed different approaches of visual feature extraction components: spatio-temporal interest point detection, trajectory formation, and visual descriptors. Thirdly, we provided a survey of different methods for global feature representation components: vocabulary generation, local feature encoding, and pooling of encoded features. Finally, we discussed various supervised machine learning methods for video classification.
Chapter 3

Visual Feature Extraction

Chapter contents

3.1 Introduction .......................................................... 29
3.2 Refined Dense Trajectories ........................................ 31
  3.2.1 Region of Interest based Sampling ......................... 31
  3.2.2 Short-window Video Stabilization .......................... 32
  3.2.3 Tracking of Interest Points ................................. 32
  3.2.4 Descriptor Computation .................................. 33
  3.2.5 Experimental Results and Analysis of the RDT .......... 34
3.3 Spatio-temporal Interest Point Detector based on Low-rank and Group-sparse Matrix Approximation ................. 41
  3.3.1 LRGS Representation of Interest Points ................ 42
  3.3.2 Approximation of LRGS Components ..................... 43
  3.3.3 Extraction of Desired STIPs .............................. 44
  3.3.4 Experimental Results and Analysis of the LRGS-STIP ... 44
3.4 Chapter Summary .................................................. 50

3.1 Introduction

Visual feature extraction is an important step in video recognition. In general, spatio-temporal interest points (STIPs) are first detected in each frame of a video. The interest points represent the key points where the visual features are most discriminative. In videos, the key points can be found where there is motion [22]–[32]. Then, local feature descriptors are calculated within a volume, either around the STIPs [22] or along trajectories formed by tracking those STIPs [23]. The
computed feature descriptors are finally used for classification after represented by a global model.

There exist many local feature extraction methods, among them dense trajectories have been proposed for temporal feature extraction [23]. Although dense trajectory method performs well for motion modeling, its classification accuracy is degraded by viewpoint changes and camera motion. Furthermore, dense sampling of spatio-temporal interest points may result in too many trajectories that add noise and increase complexity. To solve this problem, salient spatio-temporal interest points are detected in a region of interest (ROI) where there is motion. This gives a better classification accuracy by discarding unnecessary trajectories.

There are a few problems associated with the existing STIP detectors. Firstly, the detectors yield either a sparse set of STIPs, which leads to loss of information, or a dense set of STIPs, which results in additional noise and complexity [32]. Secondly, in case of dynamic background and moving camera, the detectors may find irrelevant interest points that do not belong to an actual motion. This is because the detectors are usually designed for a controlled environment with a stable background and extract STIPs without considering global motion [32]. To address these limitations, an STIP detector is designed to extract salient interest points while considering global motion.

In this chapter, firstly, we present a new method for visual feature extraction named refined dense trajectories in Section 3.2. The refined dense trajectories extract salient interest points in a region of interest (ROI) where there is motion and discards the noisy and redundant interest points. Secondly, we propose a novel spatio-temporal interest point detector based on a low-rank and group-sparse matrix approximation in Section 3.3. The proposed visual feature extraction methods are tested on different dynamic scene recognition and action recognition datasets. The classification results are presented and compared with those of some existing methods. The chapter is concluded in Section 3.4.
3.2 Refined Dense Trajectories

In this section, refined dense trajectory method for visual feature extraction is introduced. Firstly, a set of interest points is obtained in motion areas by a region of interest based sampling. Secondly, camera motion is removed through short-window video stabilization by compensating global motion. Thirdly, the interest points are tracked to form trajectories using median filtering on dense optical flow field. Finally, the local feature descriptors are computed along the trajectories and visual features are extracted.

3.2.1 Region of Interest based Sampling

The dense sampling has recently been used for feature extraction \cite{23}. The problem with dense sampling is that it usually yields too many interest points, which need to be tracked. This results in excessive trajectories that add noise and reduce the classification accuracy. To obtain salient trajectories, the proposed RDT method incorporates only interest points within a ROI that contains motion. To find the ROI, a smooth dense optical flow field is computed. Irregular and fast motion patterns can easily be tracked because of the smoothness constraints of the dense optical flow field. In the RDT method, motion detection is performed by calculating the gradient magnitude of the optical flow, yielding a gray-level image that indicates motion areas. A threshold is then applied on the gray-level image to obtain a mask, which indicates the ROI.

To find the optimum threshold, minimum error thresholding \cite{100} is used. For this purpose, the gray-level normalized histogram $p(i)$, $i = 0, 1, 2, ..., n$, is considered as an estimate of the probability density function for the foreground and background pixels. Let $P_1(\tau)$ and $P_2(\tau)$ denote, respectively, the *a priori* probabilities of foreground and background,

$$P_1(\tau) = \sum_{i=0}^{\tau} p(i), \quad \text{(3.1)}$$

$$P_2(\tau) = \sum_{i=\tau+1}^{n} p(i). \quad \text{(3.2)}$$

The following objective function $f(\tau)$ is minimized iteratively to find the minimum
3.2. Refined Dense Trajectories

Error threshold:

\[ J(\tau) = 1 + 2[P_1(\tau) \log \sigma_1(\tau) + P_2(\tau) \log \sigma_2(\tau)] - 2[P_1(\tau) \log P_1(\tau) + P_2(\tau) \log P_2(\tau)], \quad (3.3) \]

where \( \sigma_1(\tau) \) and \( \sigma_2(\tau) \) are the foreground and background variances for the threshold \( \tau \), respectively. The optimum threshold is calculated using Bayes’ minimum error rule. Briefly, this rule first selects an arbitrary threshold \( \tau \), and calculates the foreground and background means (i.e., \( \mu_1(\tau) \) and \( \mu_2(\tau) \)), variances, and \emph{a priori} probabilities. The following quadratic equation is then used to calculate the updated threshold:

\[
j^2 \left[ \frac{1}{\sigma_1^2(\tau)} - \frac{1}{\sigma_2^2(\tau)} \right] - 2j \left[ \frac{\mu_1(\tau)}{\sigma_1^2(\tau)} - \frac{\mu_2(\tau)}{\sigma_2^2(\tau)} \right] + \frac{\mu_1^2(\tau)}{\sigma_1^2(\tau)} - \frac{\mu_2^2(\tau)}{\sigma_2^2(\tau)} + 2[\log \sigma_1(\tau) - \log \sigma_2(\tau)] - 2[\log P_1(\tau) - \log P_2(\tau)] = 0. \quad (3.4)\]

The iterations terminate if the updated threshold equals the old one. For further details on Bayes’ minimum error rule, see [100].

3.2.2 Short-window Video Stabilization

In case of camera motion and dynamic background, the motion information carried by the trajectories gets corrupted. To address this problem, the video frames are spatially realigned by estimating global motion. For a frame \( t \), the subsequent frames \( t + 1 \) to \( t + L - 1 \) are aligned with it. For this purpose, SURF descriptors are computed for SURF points in two frames, and the locations of the corresponding points are retrieved by matching their SURF descriptors. An affine transformation corresponding to the matched point pairs is calculated using M-estimator sample consensus algorithm [101]. Using the estimated geometric transformation, the two frames are realigned. This stabilizes the background using the global motion and is referred to short-window video stabilization (SWVS). For a frame \( t \), the interest points are then tracked to form trajectories within the realigned frames. This yields stabilized trajectories after global motion compensation.

3.2.3 Tracking of Interest Points

The interest points need to be tracked to form trajectories. For this purpose, the same procedure used in [23] is employed. Here, we briefly describe dense
3.2. Refined Dense Trajectories

Figure 3.1: Illustration of the trajectory and descriptor computation, adapted from [23]. (a) A hand waving scene. Red points show the interest points to be tracked and green tracks represent the trajectories. A median filter is applied to the dense optical flow field to track the points. (b) Descriptors such as HOG and MBH are computed along the trajectories within a volume of size $R \times R \times L$, which is subdivided into cells of size $r_x \times r_y \times \ell$.

trajectories method for tracking of interest points. At a frame $t$, an interest point $P_t = (x_t, y_t)$ is tracked in the following frame $t + 1$, and its tracked position is smoothed using a median filter applied to the dense optical flow field $w_t = (u_t, v_t)$:

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (M * w_t)(x_t, y_t), \quad (3.5)$$

where $M$ represents a median filter kernel of 3x3 pixels, and $*$ represents the convolution operator. A trajectory is formed by concatenating tracked points in subsequent frames ($P_t, P_{t+1}, P_{t+2}, ...$), see [23] for more details. The trajectories are tracked up to only $L$ frames because they tend to drift from their point of initialization. The value of $L$ is selected based on experiments on different datasets for best classification results. Fig. 3.1(a) shows the tracking of interest points in $L$ frames.

3.2.4 Descriptor Computation

There exist many hand-crafted local descriptors to capture spatial and temporal information in videos. The local descriptors such as histogram of oriented gradient (HOG) and motion boundary histogram (MBH) have been used extensively.
3.2. Refined Dense Trajectories

HOG descriptor computes the orientation of shape at the finest level (e.g., each pixel) to capture appearance information. MBH descriptor extracts the dynamic information along the trajectories. Camera motion induces a great deal of noise in videos. Based on derivatives of optical flow, MBH descriptor is more discriminative than the optical flow, and helps with suppressing the camera motion in a simple and efficient way. MBH descriptor computes the spatial derivatives of the optical flow field for the vertical and horizontal components (i.e., MBHx and MBHy). These components encode the relative motion between pixels [44].

HOG and MBH descriptors are computed along the trajectories within a space-time volume with dimensions $R \times R \times L$, which leverages the appearance and motion information. The space-time volume is further divided into smaller grids of size $r_x \times r_y \times \ell$. Fig. 3.1(b) shows a trajectory and descriptor computation within the space-time grids along the trajectory. The proposed RDT based visual feature extraction method is summarized in Algorithm 1 (page 35).

3.2.5 Experimental Results and Analysis of the RDT

In this section, detailed results and analysis of the proposed refined dense trajectories method are presented. The proposed RDT method is analyzed on two benchmark datasets in terms of classification accuracy and computation time. The extracted features are represented by a super descriptor tensor decomposition (SDTD) model, described in Chapter 4. The classification results are obtained using an extreme learning machines classifier.

3.2.5.1 Datasets and Evaluation Protocol

The proposed RDT method is evaluated using two datasets, Maryland “in-the-wild” [103] and YUPPEN dynamic scenes [104]. These datasets contain 130 and 420 videos of natural dynamic scenes, respectively. The sample video frames from Maryland and YUPPEN datasets are shown in Fig. 3.2.

Maryland dataset was obtained from “in-the-wild” sources, e.g., amateur footage from internet, YouTube and cinematic movies. It contains dynamic scenes of thirteen classes: avalanche, boiling water, chaotic traffic, forest fire, fountain, iceberg collapse, landslide, smooth traffic, tornado, volcanic eruption, waterfall, waves, and
Algorithm 1 Visual feature extraction via refined dense trajectories

**Input:** A video sequence in gray-scale, trajectory length \( L \), dimensions of space-time volume around trajectories \( R, r_x, r_y, \ell \).

**Output:** A set of \( N \) feature vectors \( X = \{x_1, \ldots, x_N\}, x_i \in \mathbb{R}^M \), for each descriptor (HOG, MBHx, and MBHy).

1. Densely sample the interest points in video frames on a grid [23].
2. Compute a dense optical flow field using Farneback algorithm [102].
3. Calculate the magnitude gradient \( G \) of the optical flow field for each frame.
4. Obtain a binary mask by applying minimum error thresholding using (3.3) on image \( G \).
5. Apply the mask on densely sampled interest points and obtain only points in ROI at each frame.
6. Track each interest point in \( L \) frames which are realigned using SWVS for global motion compensation. Apply median filtering on the dense optical flow field for tracking [23].
7. At each frame, form \( N \) trajectories by concatenating tracked points in subsequent frames.
8. Calculate HOG, MBHx, and MBHy descriptors within a space-time volume \( R \times R \times L \), which is further divided into smaller \( r_x \times r_y \times \ell \) grids (see Fig. 3.1).

**whirlpool.** There are ten color videos for each class. The average size of the videos is \( 308 \times 417 \) (pixels) \( \times 617 \) (frames). The dataset includes noise effects like camera motion, scene cuts, and variations in viewpoint, frame rate, frame scale, and illumination.

**YUPPEN dynamic scenes** is a dataset with no camera motion. It contains scenes of fourteen classes: beach, elevator, forest fire, fountain, highway, lightning storm, ocean, railway, rushing river, sky-clouds, snowing, street, waterfall, and windmill farm. There are thirty color videos per class. The videos were collected from different sources like YouTube, Getty Images, and BBC Motion Gallery; some of the videos were made by the authors themselves [104]. The average size of the videos is \( 250 \times 370 \) (pixels) \( \times 145 \) (frames).

To be consistent with other methods tested for Maryland and YUPPEN datasets, the same evaluation protocol is adopted, that is leave-one-out evaluation. In this
3.2. Refined Dense Trajectories

(a) Maryland "in-the-wild"

(b) YUPPEN dynamic scenes

Figure 3.2: Sample video frames from Maryland “in-the-wild” \cite{103} and YUPPEN dynamic scenes \cite{104} datasets.
evaluation protocol, one sample video from a dataset is reserved for testing while the rest of the dataset is used for training. The leave-one-out process is repeated for the total number of samples in the dataset. Finally, overall classification accuracy is calculated by averaging of individual classification scores.

3.2.5.2 Experimental Method

In our implementation of the proposed RDT method, the parameter settings are as follows. The sampling step size of the interest points is 5 pixels. There are in total 8 spatial scales changing by a factor of $1/\sqrt{2}$. For the tracking of interest points, the median filter kernel is $3 \times 3$ pixels. For tracking, the algorithm by Farneback [102] is used to extract the dense optical flow field because it embeds a translational motion model between two consecutive frames. Based on experiments, the trajectory length is set to $L = 15$ for best classification results. To compute different descriptors (i.e., HOG, MBHx, and MBHy) along the trajectories, the parameter values for volume $R \times R \times L$ and spatio-temporal grid $r_x \times r_y \times \ell$ are set to $R = 32$, $r_x = 2$, $r_y = 2$, and $\ell = 3$. Using an 8-bin quantization of orientations, the final dimension of HOG, MBHx, and MBHy descriptors is 96 each. These settings give the best performance across different types of video datasets [23]. Since the trajectory and descriptor computation parts of the proposed RDT method are same as dense trajectories method, similar experimental setup is adapted for a performance comparison.

3.2.5.3 Effects of Trajectory Length

In this experiment, the effects of trajectory length $L$ on classification accuracy are analyzed. Different values of $L$ are tested and classification rates (CRs) are obtained for Maryland and YUPPEN datasets. The CR as a function of trajectory length $L$ is given in Fig. 3.3 for Maryland and YUPPEN datasets. The CRs get better for large values of $L$. The highest CRs obtained are 89.2% and 98.1% for Maryland and YUPPEN datasets, respectively, for $L = 15$. If we increase the values of $L$ further, the CRs start decreasing. This is because the trajectories start drifting from their point of initialization. The trajectory length $L$ is set to 15 in further experiments.
3.2. Refined Dense Trajectories

![Classification rate versus trajectory length L for Maryland and YUPPEN datasets.](image)

**Figure 3.3:** Classification rate versus trajectory length $L$ for Maryland and YUPPEN datasets.

### 3.2.5.4 Visual Analysis

In this subsection, the effect of removing unnecessary trajectories from the videos is analyzed. The proposed RDT method first calculates the ROI and then refines the trajectories based on that ROI. A visual comparison of the proposed refined dense trajectories with dense trajectories method in [23] is presented here. Frames of waterfall and windmill scenes from YUPPEN dynamic scenes dataset [104] are shown in Fig. 3.4(a) and (d). The trajectories computed by the method in [23] and the proposed RDT method are shown in Fig. 3.4(b) and (e), and 3.4(c) and (f), respectively. The green lines represent the motion trajectories, and the red dots represent the end points of the trajectories. Although the green trajectories move across the frames along temporal direction as shown in Fig. 3.1(a), the trajectories are projected onto single frame for simpler visualization. One can see the extra red points representing the end points of the trajectories in the static textured regions of the scene as shown in Fig. 3.4(b) and (e). The unnecessary trajectories are removed, and the salient trajectories are computed through the proposed RDT method as shown in Fig. 3.4(c) and (f).
3.2. Refined Dense Trajectories

Figure 3.4: Refined dense trajectories: (a) (d) A waterfall and windmill scene from YUPPEN dynamic scenes dataset [104]; (b) (e) Trajectories computed by the dense trajectories method in [23], the green lines represent the motion trajectories, and the red dots represent the end points of the trajectories; (c) (f) The proposed refined dense trajectories, the irrelevant trajectories (red dots in static textured region in (b) and (e)) have been removed, and only trajectories in the ROI are kept.

3.2.5.5 Evaluation based on Classification Accuracy and Computation Time

The RDT method is compared with dense trajectories method proposed by Wang et al. [23], in terms of number of extracted trajectories, classification accuracy, and computation time. The local feature descriptors obtained from the RDT and the method in [23] are processed by the super descriptor tensor decomposition model (described in Chapter 4) for classification.

A random video sequence of each category is taken from Maryland and YUPPEN datasets. The video size, number of trajectories, and total computation time of trajectories and descriptors (i.e., HOG, MBHx, and MBHy) are given in Tables 3.1 and 3.2 for the two datasets. The average number of trajectories computed for the RDT is less than that of the method in [23] (Tables 3.1 and 3.2, No. of trajectories). The complexity is reduced, and the efficiency is increased by refining
3.2. Refined Dense Trajectories

Table 3.1: Number of trajectories, total computation time (trajectories and descriptors), and CR ± std (average and per-category) of the proposed RDT and the method of Wang et al. [23], for Maryland “in-the-wild” dataset. For a better comparison between the two methods, the global motion compensation using short-window video stabilization is not added.

<table>
<thead>
<tr>
<th>Categories</th>
<th>Video size (height x width x frames)</th>
<th>No. of trajectories</th>
<th>Computation time (s)</th>
<th>CR ± std (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>RDT [23]</td>
<td>RDT</td>
</tr>
<tr>
<td>Avalanche</td>
<td>384 x 288 x 239</td>
<td>98643</td>
<td>48672</td>
<td>70.39</td>
</tr>
<tr>
<td>Boiling water</td>
<td>352 x 288 x 251</td>
<td>89925</td>
<td>60911</td>
<td>66.08</td>
</tr>
<tr>
<td>Chaotic traffic</td>
<td>320 x 240 x 2044</td>
<td>480033</td>
<td>145507</td>
<td>416.18</td>
</tr>
<tr>
<td>Forest fire</td>
<td>320 x 240 x 934</td>
<td>136856</td>
<td>84012</td>
<td>172.15</td>
</tr>
<tr>
<td>Fountain</td>
<td>320 x 240 x 126</td>
<td>26008</td>
<td>8398</td>
<td>30.76</td>
</tr>
<tr>
<td>Iceberg collapse</td>
<td>464 x 348 x 1217</td>
<td>35206</td>
<td>1705</td>
<td>431.05</td>
</tr>
<tr>
<td>Landslide</td>
<td>320 x 240 x 677</td>
<td>254313</td>
<td>93640</td>
<td>162.48</td>
</tr>
<tr>
<td>Smooth traffic</td>
<td>320 x 240 x 1218</td>
<td>294837</td>
<td>82350</td>
<td>270.75</td>
</tr>
<tr>
<td>Tornado</td>
<td>320 x 240 x 177</td>
<td>19171</td>
<td>6409</td>
<td>29.35</td>
</tr>
<tr>
<td>Volcano eruption</td>
<td>480 x 320 x 2000</td>
<td>9149</td>
<td>6910</td>
<td>601.37</td>
</tr>
<tr>
<td>Waterfall</td>
<td>450 x 360 x 745</td>
<td>669249</td>
<td>192462</td>
<td>365.04</td>
</tr>
<tr>
<td>Waves</td>
<td>480 x 360 x 321</td>
<td>156148</td>
<td>33000</td>
<td>135.12</td>
</tr>
<tr>
<td>Whirlpool</td>
<td>480 x 360 x 1379</td>
<td>565888</td>
<td>347346</td>
<td>619.95</td>
</tr>
<tr>
<td>Average</td>
<td>385 x 289 x 871</td>
<td>218109</td>
<td>85486</td>
<td>259.30</td>
</tr>
</tbody>
</table>

The run-time is obtained on a desktop computer with 3.50 GHz Intel i7 CPU. The average computation time for the trajectories and descriptors calculation for the RDT is 19.1% and 16.0% less than that of the method in [23] (Tables 3.1 and 3.2, Computation time).

A comparison in terms of classification accuracy is presented for the RDT and the method in [23]. The CRs and standard deviation (std) for the two methods are presented in Tables 3.1 and 3.2. For a better comparison between the two methods, the global motion compensation using short-window video stabilization is not added. The method in [23] achieves average CRs of 80.00% and 96.67% on Maryland and YUPPEN datasets, respectively. In comparison, the proposed RDT method achieves average CRs of 84.62% and 98.10% for Maryland and YUPPEN datasets, respectively. The RDT method achieves higher CRs than the method in [23] for the two datasets.

The statistical significance of the difference between CRs of the RDT and the method in [23] is assessed using the Friedman’s test [105] (described in Section 5.2.2). A p-value is obtained which indicates whether there is a significant difference between the CRs of the two methods. For a significance level of
3.3 Spatio-temporal Interest Point Detector based on Low-rank and Group-sparse Matrix Approximation

Table 3.2: Number of trajectories, total computation time (trajectories and descriptors), and CR ± std (average and per-category) of the proposed RDT and method of Wang et al. [23], for YUPPEN dynamic scenes dataset.

<table>
<thead>
<tr>
<th>Categories</th>
<th>Video size (height x width x frames)</th>
<th>No. of trajectories</th>
<th>Computation time (s)</th>
<th>CR ± std (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[23]</td>
<td>RDT [23]</td>
<td>RDT</td>
<td>[23]</td>
</tr>
<tr>
<td>Beach</td>
<td>480 x 270 x 150</td>
<td>18775</td>
<td>866</td>
<td>43.11</td>
</tr>
<tr>
<td></td>
<td>RDT</td>
<td>35.82</td>
<td>96.67</td>
<td>96.67</td>
</tr>
<tr>
<td>Elevator</td>
<td>418 x 270 x 150</td>
<td>901</td>
<td>275</td>
<td>33.32</td>
</tr>
<tr>
<td></td>
<td>RDT</td>
<td>31.02</td>
<td>93.33</td>
<td>100.00</td>
</tr>
<tr>
<td>Forest fire</td>
<td>320 x 217 x 81</td>
<td>12443</td>
<td>8936</td>
<td>14.44</td>
</tr>
<tr>
<td></td>
<td>RDT</td>
<td>14.44</td>
<td>93.33</td>
<td>100.00</td>
</tr>
<tr>
<td>Fountain</td>
<td>320 x 293 x 150</td>
<td>28211</td>
<td>20059</td>
<td>38.52</td>
</tr>
<tr>
<td></td>
<td>RDT</td>
<td>34.00</td>
<td>93.33</td>
<td>93.33</td>
</tr>
<tr>
<td>Highway</td>
<td>320 x 226 x 150</td>
<td>36092</td>
<td>3075</td>
<td>28.95</td>
</tr>
<tr>
<td></td>
<td>RDT</td>
<td>21.64</td>
<td>96.67</td>
<td>100.00</td>
</tr>
<tr>
<td>Lightning storm</td>
<td>320 x 212 x 150</td>
<td>14093</td>
<td>5043</td>
<td>25.21</td>
</tr>
<tr>
<td></td>
<td>RDT</td>
<td>22.46</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Ocean</td>
<td>480 x 270 x 150</td>
<td>8464</td>
<td>6070</td>
<td>45.01</td>
</tr>
<tr>
<td></td>
<td>RDT</td>
<td>45.13</td>
<td>93.33</td>
<td>100.00</td>
</tr>
<tr>
<td>Railway</td>
<td>320 x 252 x 117</td>
<td>14858</td>
<td>6274</td>
<td>22.34</td>
</tr>
<tr>
<td></td>
<td>RDT</td>
<td>20.30</td>
<td>96.67</td>
<td>96.67</td>
</tr>
<tr>
<td>Rushing river</td>
<td>320 x 226 x 150</td>
<td>16424</td>
<td>10774</td>
<td>29.86</td>
</tr>
<tr>
<td></td>
<td>RDT</td>
<td>23.33</td>
<td>93.33</td>
<td>96.67</td>
</tr>
<tr>
<td>Sky-clouds</td>
<td>384 x 218 x 150</td>
<td>36689</td>
<td>19405</td>
<td>29.94</td>
</tr>
<tr>
<td></td>
<td>RDT</td>
<td>25.45</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Snow</td>
<td>480 x 270 x 150</td>
<td>156492</td>
<td>86599</td>
<td>84.82</td>
</tr>
<tr>
<td></td>
<td>RDT</td>
<td>66.39</td>
<td>96.67</td>
<td>96.67</td>
</tr>
<tr>
<td>Street</td>
<td>320 x 265 x 150</td>
<td>15078</td>
<td>4443</td>
<td>32.12</td>
</tr>
<tr>
<td></td>
<td>RDT</td>
<td>27.51</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Waterfall</td>
<td>340 x 231 x 150</td>
<td>40433</td>
<td>10465</td>
<td>35.45</td>
</tr>
<tr>
<td></td>
<td>RDT</td>
<td>24.48</td>
<td>96.67</td>
<td>93.33</td>
</tr>
<tr>
<td>Windmill farm</td>
<td>382 x 270 x 150</td>
<td>37298</td>
<td>8362</td>
<td>41.74</td>
</tr>
<tr>
<td></td>
<td>RDT</td>
<td>32.89</td>
<td>96.67</td>
<td>100.00</td>
</tr>
<tr>
<td>Average</td>
<td>372 x 250 x 143</td>
<td>31161</td>
<td>13617</td>
<td>36.13</td>
</tr>
<tr>
<td></td>
<td>RDT</td>
<td>30.34</td>
<td>96.67±0.88</td>
<td>98.10±0.67</td>
</tr>
</tbody>
</table>

5%, if the $p$-value $\leq 0.05$, then the difference between CRs of the two methods is statistically significant, otherwise it is not. The $p$-values are calculated by comparing the RDT and the method in [23] using the per-category CRs in Tables 3.1 and 3.2. The $p$-values are 0.10 and 0.11 (larger than 0.05) for Maryland and YUPPEN datasets, respectively. Although the difference between the CRs of the two methods is not significant, the RDT method achieves higher CRs than the method in [23].

3.3 Spatio-temporal Interest Point Detector based on Low-rank and Group-sparse Matrix Approximation

In this section, a new spatio-temporal interest point detector based on a low-rank and group-sparse matrix approximation is proposed. The spatial interest points (SIPs) are detected in a video using existing corner, edge, and blob feature detectors. A short-window video stabilization is integrated in the detector for global motion compensation. The desired STIPs are then detected based on a low-rank and group-sparse (LRGS) matrix approximation. The proposed LRGS based STIP detector is explained next.
3.3. Spatio-temporal Interest Point Detector based on Low-rank and Group-sparse Matrix Approximation

3.3.1 LRGS Representation of Interest Points

For a video, the SIPs are detected in each frame, using FAST corners [106], Canny edges [107], and SURF features [47]. Each frame is first scanned for the detected SIPs, using row-wise, column-wise, or zig-zag scanning. Let $S = \{(x, y, t)\}$, $i = 1, 2, ..., S$, be a set of all the SIPs detected in the video. For the $i$th SIP, an $L$-dimensional column vector $v_i$, $i = 1, 2, ..., S$, is formed, which contains the pixel values in the $L$ frames, $v_i = [I(x, y, t + j)]$, $j = 0, 1, ..., L - 1$. To compensate for global camera motion, the frames $j = 1, 2, ..., L - 1$ are realigned with the frame $j = 0$ (process described in Section 3.2.2). For $S$ number of SIPs, a matrix $\Phi \in \mathbb{R}^{L \times S}$ is formed using the SIP vectors $v_i$, as columns,

$$\Phi = [v_1, v_2, ..., v_S]. \quad (3.6)$$

This formation of matrix $\Phi$ is unique and clearly different from the many approaches in video analysis, where the video frames are arranged as columns of matrix $\Phi$ [108].

Each column in matrix $\Phi$ corresponds to an SIP vector. A vector having constant pixel values in $L$ frames belongs to static background, and a vector having varying pixel values in $L$ frames belongs to moving foreground. The SIP vectors that belong to the moving foreground are the desired STIPs. The background SIP vectors in $\Phi$ can be separated as a low-rank component of matrix $\Phi$, whereas the foreground SIP vectors can be separated as outliers or as a sparse component of matrix $\Phi$. We consider the decomposition of $\Phi$ into low-rank and group-sparse components plus noise as follows:

$$\Phi = B + F + N, \quad (3.7)$$

where $B$ is low-rank background matrix, $F$ is group-sparse foreground matrix, and $N$ represents additive noise. To detect an SIP as a desired STIP, the corresponding column/group in $\Phi$, as a whole needs to be approximated as sparse in $F$. This is referred to as group-sparsity.
3.3. Spatio-temporal Interest Point Detector based on Low-rank and
group-sparse Matrix Approximation

3.3.2 Approximation of LRGS Components

Based on the low-rank and group-sparsity constraints, the following objective function is to be minimized:

$$
\min_{B,F} \|B\|_* + \lambda \|F\|_{2,1}, \text{ s.t. } \|\Phi - B - F\|_F^2 \leq \epsilon,
$$

where $$\|\cdot\|_*$$ is the nuclear norm, $$\|\cdot\|_F$$ is the Frobenius norm, $$\|\cdot\|_{2,1}$$ is a mixed $$\ell_{2,1}$$-norm, and $$\lambda$$ is a regularization parameter to control the sparsity in $$F$$.

The optimization problem in (3.8) is solved by inexact augmented Lagrangian multiplier (IALM) method [109]. The augmented Lagrangian formulation of (3.8) is given as

$$
L(B,F,Y,\mu) = \|B\|_* + \lambda \|F\|_{2,1} + \langle Y, \Phi - B - F \rangle + \frac{\mu}{2} \|\Phi - B - F\|_F^2,
$$

where $$Y$$ is the Lagrange multiplier and $$\mu$$ is a positive scalar. The problem in (3.9) is decoupled into the following sub-problems:

$$
B_{k+1} = \min_B L(B,F_k,Y_k,\mu_k)
= \min_B \left( \|B\|_* + \lambda \|F_k\|_{2,1} + \langle Y_k, \Phi - B_k - F_k \rangle + \frac{\mu_k}{2} \|\Phi - B_k - F_k\|_F^2 \right) \tag{3.10}
= \min_B \left( \|B\|_* + \frac{\mu_k}{2} (\Phi - B_k - F_k + \mu_k^{-1} Y_k) \right),
$$

$$
F_{k+1} = \min_F L(B_{k+1},F,Y_k,\mu_k)
= \min_F \left( \lambda \|F\|_{2,1} + \langle Y_k, \Phi - B_{k+1} - F \rangle + \frac{\mu_k}{2} \|\Phi - B_{k+1} - F\|_F^2 \right) \tag{3.11}
= \min_F \left( \lambda \|F\|_{2,1} + \frac{\mu_k}{2} (\Phi - B_{k+1} - F + \mu_k^{-1} Y_k) \right),
$$

$$
Y_{k+1} = Y_k + \mu_k (\Phi - B_{k+1} - F_{k+1}). \tag{3.12}
$$

The above sub-problems are solved alternately. To solve (3.10), first, the singular value decomposition (SVD) of $$(\Phi - F_k + \mu_k^{-1} Y_k)$$ is computed,

$$
(\Phi - F_k + \mu_k^{-1} Y_k) = U \Sigma V^T,
$$

where $$U$$ and $$V$$ are matrices of left and right singular vectors, respectively, and $$\Sigma$$ is a diagonal matrix of singular values $$\sigma_l$$, $$l = 1, 2, ..., L$$. Matrix $$B_{k+1}$$ is then calculated by singular value soft-thresholding as

$$
B_{k+1} = U \Delta(\Sigma, \mu_k^{-1}) V^T, \tag{3.14}
$$
where $\Delta(S, \mu^{-1}_k) = \text{diag}(\max(\sigma_l - \mu^{-1}_k, 0))$ represents the shrinkage operation. This shrinks the singular values towards zero to minimize the rank of matrix $B$.

To solve (3.11), a soft-thresholding operation is applied. Let $r_i$ denote the $i$th column of $(\Phi - B_{k+1} + \mu^{-1}_k Y_k)$. The $i$th column of matrix $F_{k+1}$, $f_i$, is obtained as

$$f_i = r_i \left\{ \max \left(0, 1 - \lambda \mu^{-1}_k / \|r_i\|_2 \right) \right\}.$$

At each iteration, parameter $\mu$ is updated as $\mu_{k+1} = \rho \mu_k$, where $\rho > 1$. This parameter is not updated further if it reaches a predefined limit $\mu_{\text{max}}$. For convergence, an error $\|\Phi - B_{k+1} - F_{k+1}\|_F^2 / \|\Phi\|_F^2$ is computed at each iteration. The alternating minimization terminates if the error becomes less than a threshold $\tau$ or the number of iterations reaches a maximum limit $Z$.

### 3.3.3 Extraction of Desired STIPs

Each column in $F_{k+1}$ corresponds to an SIP. If a column in $F_{k+1}$ is zero, the corresponding SIP belongs to the static background, and if the column is non-zero, the corresponding SIP belongs to the moving foreground. The extracted foreground SIPs are the desired STIPs. The STIP detection process is illustrated in Fig. 3.5. The input video shows a moving train, the SIPs are detected using FAST corners, Canny edges, and SURF features, and the STIPs are detected via the proposed LRGS-STIP detector. The proposed LRGS-STIP detector based visual feature extraction method is summarized in Algorithm 2 (page 46).

### 3.3.4 Experimental Results and Analysis of the LRGS-STIP

In this section, detailed results and analysis of the proposed LRGS-STIP detector are presented. In our recognition system, the visual features are extracted by the LRGS-STIP method. The feature descriptors are represented by the super descriptor tensor decomposition model, described in Chapter 4. An extreme learning machine classifier is used to obtain the classification results in the end.

#### 3.3.4.1 Datasets and Evaluation Protocol

The proposed LRGS-STIP detector is tested on four action recognition datasets: KTH [110], UCF [111], YouTube [112], and MSR-I [113].
3.3. Spatio-temporal Interest Point Detector based on Low-rank and Group-sparse Matrix Approximation

Figure 3.5: STIP detection through the proposed LRGS-STIP detector. Input video shows a scene of a moving train. The SIPs are detected using FAST corners, SURF features, and Canny edges. The desired STIPs are then detected via the proposed LRGS-STIP detector.

KTH dataset contains 600 videos of 6 different actions: walking, jogging, running, boxing, clapping, and waving. These actions are performed by 25 different actors in a controlled environment with static and homogenous background. For evaluation, the same protocol used in [110] is adopted here; that is, videos from 9 actors are used for testing and the rest for training purpose.

UCF dataset contains 150 videos of 10 different sport actions: diving, golf swing, kicking, lifting, riding horse, running, skate boarding, swing-bench, swing-side, and walking. The dataset involves dynamic and cluttered backgrounds with multiple view-points. For performance evaluation phase with UCF dataset, the leave-one-out cross-validation is used as in [111].

YouTube dataset consists of 1168 videos of 11 actions: basketball shooting, volleyball spiking, trampoline jumping, soccer juggling, horseback riding, cycling, diving, swinging, golf swinging, tennis swinging, and walking. The videos are a mix of static, moving, and cluttered backgrounds with variations in object scale, view-point, and illumination. For performance evaluation phase with YouTube dataset, the leave-one-out cross-validation is used as in [112].
Algorithm 2 Visual feature extraction via the LRGS-STIP detector

**Input:** A video sequence in gray-scale, trajectory length $L$, dimensions of space-time volume around trajectories $R, r_x, r_y, \ell$, LRGS parameters $\lambda, \rho, \mu_{\text{max}}, \tau, Z$.

**Output:** A set of $N$ feature vectors $X = \{x_1, ..., x_N\}, x_i \in \mathbb{R}^M$, for each descriptor.

1. Detect a set of SIPs, $S = \{(x, y, t)\}, i = 1, ..., S$, using FAST corners, Canny edges, and SURF features.

2. Form a column vector $v_i \in \mathbb{R}^L$ for the $i$th SIP, which contains the pixel values in the $L$ frames, $v_i = [I(x, y, t + j)], j = 0, 1, ..., L - 1$. The frames $j = 1, ..., L - 1$, are realigned with the frame $j = 0$ for global motion compensation.

3. Form a matrix $\Phi \in \mathbb{R}^{L \times S}$ using the SIP vectors $v_i$ as columns, as in (3.6), and decompose the matrix $\Phi$ into a low-rank matrix $B$ and a group-sparse matrix $F$, as in (3.7), by solving the sub-problems in (3.10) and (3.11).

4. Alternately solve the sub-problems (3.10) and (3.11) to estimate $B$ and $F$, through soft-thresholding using (3.14) and (3.15), respectively.

5. At each iteration, update the Lagrangian multiplier $Y$ in (3.12), where $\mu_{k+1} = \rho \mu_k, \rho > 1$, and $\mu < \mu_{\text{max}}$, and terminate the alternating minimization if the error $\|\Phi - B_{k+1} - F_{k+1}\|_F^2 / \|\Phi\|_F^2 < \tau$ or the number of iterations reaches $Z$.

6. Obtain the desired STIPs by extracting SIPs corresponding to the non-zero columns of $F$.

7. Compute a dense optical flow field using Farneback algorithm [102], then track each STIP in the SWVS realigned $L$ frames by applying median filtering on the dense optical flow field [23].

8. Form $N$ trajectories by concatenating tracked points in subsequent frames. Calculate HOG, MBHx, and MBHy descriptors along the trajectories within a space-time volume $R \times R \times L$, divided into $r_x \times r_y \times \ell$ grids (see Fig. 3.1).

MSR-I dataset contains 63 instances of 3 actions: hand-clapping, hand-waving, and boxing. The videos contain indoor and outdoor scenes with cluttered and moving backgrounds. For evaluation, the dataset provides ground-truth bounding boxes to estimate what percentages of STIPs belong to actions and background. Some sample video frames from KTH, UCF, YouTube, and MSR-I datasets are shown in Fig. 3.6.

3.3.4.2 Experimental Method

For visual feature extraction using the LRGS-STIP detector, all experiments are conducted with the same parameter settings (except for the regularization param-
3.3. Spatio-temporal Interest Point Detector based on Low-rank and Group-sparse Matrix Approximation

Figure 3.6: Some sample video frames from action recognition datasets: KTH [110], UCF [111], YouTube [112], and MSR-I [113].
3.3. Spatio-temporal Interest Point Detector based on Low-rank and Group-sparse Matrix Approximation

The length of the vectors \( v_i \) forming the columns of \( \Phi \) is set to \( L = 15 \). In the alternating minimization of the augmented Lagrangian sub-problems, the following parameters and initial values are used: \( Y_0 = \Phi \), \( F_0 = 0 \), \( \mu_0 = 1.25/\|\Phi\|_F^2 \), \( \mu_{\text{max}} = \mu_0 \times 10^7 \), and \( \rho = 1.5 \). For convergence, the error threshold \( \tau = 1 \times 10^{-5} \) and the maximum number of iterations \( Z = 100 \). These settings work for all the datasets used. To control sparsity, parameter \( \lambda \) is tuned manually for each dataset separately: it is set to 0.009 for KTH, 0.015 for UCF, and 0.020 for YouTube dataset.

3.3.4.3 Evaluation based on Detection Ratio

The proposed LRGS-STIP detector is evaluated using MSR-I dataset by calculating ratio of valid STIPs. MSR-I dataset provides ground-truth bounding boxes for actors performing actions. A score is estimated for the number of STIPs for actions in comparison to the background STIPs. Recently, it was reported in [114] that only 18.73% of STIPs detected by Harris3D [22] correspond to the actions, while the rest belong to the background. A STIP is considered valid if it falls inside a bounding box. The proposed LRGS-STIP detector is evaluated in similar way. The ratio of valid STIPs detected by the LRGS-STIP and other detectors using MSR-I dataset is given in Table 3.3. The LRGS-STIP detects 78.94% STIPs for the actions, which is highest in comparison with other detectors. The STIP detection ratio of the method in [28] is 76.21%, which is the second highest after the LRGS-STIP detector.

Table 3.3: Average ratio ± std (in percent) of valid STIPs detected by the proposed LRGS-STIP and other detectors for MSR-I dataset. The ratio is the number of STIPs detected for actions divided by the total number of detected STIPs.

<table>
<thead>
<tr>
<th>Detector</th>
<th>Ratio ± Std (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRGS-STIP</td>
<td>78.94 ± 5.1</td>
</tr>
<tr>
<td>Selective STIPs</td>
<td>76.21 ± 5.4</td>
</tr>
<tr>
<td>Harris3D [22]</td>
<td>18.73 ± 4.9</td>
</tr>
<tr>
<td>Cuboids [26]</td>
<td>21.36 ± 5.2</td>
</tr>
</tbody>
</table>

3.3.4.4 Evaluation based on Classification Accuracy

In this experiment, the proposed LRGS-STIP detector is compared with Harris3D [22], cuboids [26], and dense sampling [23] methods in terms of classification accuracy. The trajectories and descriptors are computed from the STIPs found...
by their respective detectors. The descriptors are encoded using super descriptor vector (SDV) [115] (described in Section 4.2), and the resultant features are concatenated, to obtain a single vector for classification.

The classification results of the LRGS-STIP, Harris3D, cuboids, and dense sampling methods are presented in Table 3.4. The global motion compensation using SWVS is not added in this experiment. For KTH dataset, the LRGS-STIP achieves the highest CR, whereas the cuboids yields the lowest CR among all the detectors. The performance of the LRGS-STIP, dense sampling, and Harris3D detectors is almost similar. This is because KTH has smooth and stable backgrounds with single actor. For UCF dataset, the LRGS-STIP achieves the highest CR and cuboids has the lowest CR. For YouTube dataset, dense sampling achieves the highest CR and cuboids yields the lowest CR. The CRs of the LRGS-STIP and dense sampling are almost similar across the three datasets, whereas the LRGS-STIP yields considerable improvement over Harris3D and cuboids detectors for UCF and YouTube datasets.

Table 3.4: Average CR ± std (in percent) of the proposed LRGS-STIP and other STIP detectors for KTH, UCF, and YouTube datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>LRGS-STIP</th>
<th>Dense Sampling [23]</th>
<th>Harris3D [22]</th>
<th>Cuboids [26]</th>
</tr>
</thead>
<tbody>
<tr>
<td>KTH</td>
<td>96.0 ± 0.8</td>
<td>95.8 ± 0.8</td>
<td>95.2 ± 0.9</td>
<td>88.0 ± 1.3</td>
</tr>
<tr>
<td>UCF</td>
<td>90.7 ± 2.4</td>
<td>90.0 ± 2.4</td>
<td>84.7 ± 2.9</td>
<td>80.0 ± 3.3</td>
</tr>
<tr>
<td>YouTube</td>
<td>87.0 ± 1.0</td>
<td>87.9 ± 1.0</td>
<td>81.8 ± 1.1</td>
<td>79.2 ± 1.2</td>
</tr>
</tbody>
</table>

In the next experiment, the global motion compensation using the SWVS is added to all the STIP detectors (LRGS-STIP, dense sampling, Harris3D, and cuboids). The classification results for UCF and YouTube datasets (which involve dynamic background and camera motion) are presented in Table 3.5. KTH dataset is not used for this experiment because it contains stable backgrounds with no camera motions. The results indicate that the performance of all the detectors improves by adding global motion compensation. The LRGS-STIP achieves the highest CRs for the two datasets, whereas the cuboids method yields the lowest CRs. Although the CR of dense sampling is nearly similar to the proposed LRGS-STIP detector, the proposed detector still outperforms dense sampling.
Table 3.5: Average CR ± std (in percent) of the proposed LRGS-STIP and other STIP detectors for UCF and YouTube datasets. The global motion compensation is added for all the detectors.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>LRGS-STIP</th>
<th>Dense Sampling [23]</th>
<th>Harris3D [22]</th>
<th>Cuboids [26]</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCF</td>
<td>92.0 ± 2.2</td>
<td>91.3 ± 2.3</td>
<td>86.7 ± 2.8</td>
<td>82.7 ± 3.1</td>
</tr>
<tr>
<td>YouTube</td>
<td>88.8 ± 0.9</td>
<td>88.5 ± 0.9</td>
<td>84.0 ± 1.1</td>
<td>81.8 ± 1.1</td>
</tr>
</tbody>
</table>

### 3.4 Chapter Summary

In this chapter, we presented the refined dense trajectories method and a low-rank and group-sparse matrix approximation based STIP detector. The proposed refined dense trajectories method excludes irrelevant trajectories by incorporating only those that are present in the region of interest. The experimental results show that the RDT outperforms the widely used dense trajectories method in terms of classification accuracy and computation time. The proposed LRGS-STIP detector extracts a salient set of interest points by taking into account the camera motion and dynamic background. The experiments show that the proposed LRGS-STIP detector outperforms Harris3D, cuboids, and dense sampling methods in terms of classification accuracy.
Chapter 4

Super Descriptor Tensor Decomposition

Chapter contents

4.1 Introduction ............................................. 52
4.2 Encoding Local Feature Descriptors ............... 52
   4.2.1 Gaussian Mixture Modeling ............... 53
   4.2.2 Sparse Dictionary Learning ............... 54
   4.2.3 Feature Encoding ......................... 54
4.3 Tensor Decomposition and Feature Selection ........ 55
   4.3.1 TUCKER-3 Tensor Decomposition .......... 55
   4.3.2 Higher-order Orthogonal Interactions .... 57
   4.3.3 Fisher Ranking for Feature Selection ... 58
4.4 Experimental Results and Analysis ............... 59
   4.4.1 Datasets and Evaluation Protocol ........ 59
   4.4.2 Experimental Setup ....................... 60
   4.4.3 Performance of Different Classifiers .... 62
   4.4.4 Effects of Dictionary Size in Feature Encoding .... 63
   4.4.5 Evaluation of Tensor Decomposition Algorithms .... 64
   4.4.6 Effects of Fitness Threshold $\theta$ during Tensor Decomposition .... 65
   4.4.7 Analysis of Feature Selection after Tensor Decomposition .... 66
   4.4.8 Analysis of Different Configurations of the SDTD Model .... 67
   4.4.9 Comparison of the SDTD model with other Feature Representation Models .... 69
4.5 Chapter Summary ....................................... 72
4.1 Introduction

Local features extracted using multiple descriptors are usually represented by the traditional bag-of-words (BoW) model. More recent super vector based methods, such as super vector coding (SVC) [59], Fisher vector (FV) [61], and vector of locally aggregated descriptors (VLAD) [60], have shown to outperform the traditional BoW models. These methods aggregate high-order statistics and yield very high-dimensional representations. In addition, the encoded features from multiple feature descriptors are usually concatenated to get a large single vector for an entire video sequence. However, rather than forming a large single vector, it is more efficient to encode data from multiple descriptors in multi-dimensional arrays (i.e., tensors). Furthermore, multiple feature descriptors and multiple modalities generate massive amount of data, which increases complexity and limits many practical applications. To solve these problems, a tensor decomposition (e.g., PARAFAC and TUCKER) followed by a feature selection is applied. The TUCKER decomposition performs model reduction and feature extraction for high-dimensional tensors. The PARAFAC model is a special case of TUCKER model when all the elements in a tensor are non-zero, for details see [116]. Tensor decomposition provides an efficient tool for model reduction by capturing multi-linear structures in high-order large-scale data.

In this chapter, a super descriptor tensor decomposition (SDTD) model is proposed to represent local features that are extracted from multiple descriptors and modalities. In the proposed tensor model, the local feature descriptors are first encoded through super descriptor vector coding, described in Section 4.2. The encoded features are then arranged in the form of tensors, and discriminative features for classification are obtained through the TUCKER-3 decomposition followed by Fisher ranking, described in Section 4.3. In the end, a detailed experimental evaluation is presented in Section 4.4.

4.2 Encoding Local Feature Descriptors

To obtain a meaningful and discriminative representation, the local feature descriptors (MFCC, HOG, MBHx, and MBHy) are encoded by a coding technique
4.2. Encoding Local Feature Descriptors

before arranging the descriptors in the form of tensors. Super descriptor vector (SDV) coding [115] is adopted for this purpose. For each descriptor (MFCC, HOG, MBHx, and MBHy), a set of feature vectors \( X = \{ x_1, ..., x_N \} \), \( x_i \in \mathbb{R}^M \), is obtained from a video sequence. For MFCC descriptor, \( x_i's \) represent the MFCC feature vectors extracted from \( N = N_a \) audio frames. For the visual descriptors (HOG, MBHx, and MBHy), \( x_i's \) represent feature vectors computed by their respective descriptor along \( N = N_t \) trajectories.

The feature vector set \( X \) for each local descriptor is encoded separately using SDV coding. Firstly, the local feature descriptors are modeled using a Gaussian mixture model (GMM). Secondly, the parameters of the GMM are estimated through a sparse coding based dictionary learning method. Finally, the coefficient-weighted differences between local descriptors and codewords are integrated to obtain the encoded features.

### 4.2.1 Gaussian Mixture Modeling

The distribution of the feature vectors, \( x_i \), is represented by a GMM as follows:

\[
p(x_i) = \sum_{k=1}^{K} w_k \mathcal{N}(x_i; \mu_k, \sigma_k), \quad (4.1)
\]

where \( w_k \) is the mixture weight of the \( k \)th component density \( \mathcal{N}(x_i; \mu_k, \sigma_k) \). The mixture weight \( w_k \) corresponds to the prior probability that \( x_i \) is generated by component \( k \). The \( k \)th component density \( \mathcal{N}(x_i; \mu_k, \sigma_k) \) is a Gaussian probability density function with mean vector \( \mu_k \) and covariance matrix \( \sigma_k \). The probability density function in (4.1) models the generation process of \( x_i \).

The feature vectors \( x_i, i = 1, 2, ..., N \), are encoded by computing weighted differences between feature vectors \( x_i \) and mean vectors \( \mu_k \). For this purpose, the gradient of the log-likelihood of the function in (4.1) with respect to its parameters (i.e., mean) is calculated [117] (see Appendix A.1 for the derivation):

\[
\frac{\partial}{\partial \mu_k} \ln p(x_i) = p_k^+(x_i) \sigma_k^{-1}(x_i - \mu_k), \quad (4.2)
\]

where \( p_k^+(x_i) \) is the posterior. Equation (4.2) yields the required weighted differences; it does not require the computation of mixture weights \( w_k \).
4.2. Encoding Local Feature Descriptors

4.2.2 Sparse Dictionary Learning

The mean $\mu_{k}$ and the posterior $p_{k}^{+}(x_{i})$ in (4.2) are estimated through sparse dictionary learning. Let $\mathcal{D} = \{d_{1},...,d_{K}\}$ be a dictionary with $K$ codewords, $d_{k} \in \mathbb{R}^{M}$. There exist many sparse coding dictionary learning methods in literature [118]–[124]. The sparse coding approximates $x_{i}$ as a linear combination of a limited number of codewords. The following $\ell_{1}$ sparse coding problem is solved as in [118], [120], and [123] for the dictionary learning,

$$\min_{\mathcal{D},\mathbf{a}} \frac{1}{N} \sum_{i=1}^{N} \left( \frac{1}{2} \|x_{i} - \mathcal{D}\mathbf{a}_{i}\|_{2}^{2} + \eta \|\mathbf{a}_{i}\|_{1} \right), \text{ subject to } \mathbf{d}_{k}^{T}\mathbf{d}_{k} \leq 1, \, k = 1,\ldots,K, \tag{4.3}$$

where $\eta$ is the sparsity-inducing regularization parameter that controls the number of non-zero sparse coding coefficients in $\mathbf{a}_{i}$. The codewords $\mathbf{d}_{k}$ are constrained to have an $\ell_{2}$-norm less than or equal to 1, to prevent $\mathcal{D}$ from being arbitrarily large. It has been found empirically that the $\ell_{1}$-norm is better than the $\ell_{0}$-norm and more robust to irrelevant features [118], [120].

4.2.3 Feature Encoding

The feature vectors $x_{i} \in \mathbb{R}^{M}$ can be represented by GMM and sparse coding. A few approximations can be made to relate the two representations. Firstly, in the GMM representation, the mean vectors $\mu_{k}, \, k = 1,2,\ldots,K$, represent the centers of $K$ Gaussian components (codewords in a GMM based dictionary). In the sparse coding representation, the learned dictionary $\mathcal{D} = \{d_{1},...,d_{K}\}$ contains the centers $d_{k}, \, k = 1,2,\ldots,K$, of $K$ clusters. In the two representations, $\mu_{k}$ and $d_{k}$ representing the centers of the $k$th Gaussian component and $k$th cluster, respectively, are the codewords of a learned dictionary. Thus, an approximation can be made such that $\mu_{k} = d_{k}$. Secondly, in the GMM representation, the posterior $p_{k}^{+}(x_{i})$ is soft-assignment of $x_{i}$ to the $k$th Gaussian component (or center of the $k$th Gaussian) [61]. In the sparse coding representation, the coefficients $a_{i}^{k}$ represent the assignment of $x_{i}$ to the $k$th codeword [124]. In other words, we can write $p_{k}^{+}(x_{i}) = a_{i}^{k}$. Thirdly, the covariance matrix is assumed to be isotropic to avoid over-fitting and computational complexity, $a_{k} = \sigma^{2}I$, the same assumption is made also in [125] and [126].

54
4.3. Tensor Decomposition and Feature Selection

Based on the above approximations, the right-hand side of (4.2) can be simplified to $a_i^k(x_i - d_k)$, which are the required coefficient-weighted differences between local descriptors and codewords. Finally, for each codeword, average pooling is used to aggregate the weighted difference vectors:

$$u_k = \frac{1}{N} \sum_{i=1}^{N} a_i^k(x_i - d_k).$$

(4.4)

The resultant vectors $u_k, k = 1, 2, ..., K$, yield the encoded features.

4.3 Tensor Decomposition and Feature Selection

The encoded features from multiple descriptors (MFCC, HOG, MBHx, and MBHy) are combined using tensors. Simple concatenation of different types of features into a large single vector for classification destroys the spatio-temporal structure of the features. This concatenation also includes redundant and noisy features which can affect the classification accuracy and increase model complexity.

One can benefit from the tensor model to solve the above problems. Firstly, it is more efficient to represent different types of features in the form of multidimensional arrays than simply concatenating the features. Tensors provide a natural way to integrate features to retain the spatio-temporal structure among different type of features. Secondly, tensor decomposition followed by feature selection will discard the noisy and redundant features to improve classification accuracy and reduce model complexity. This is helpful because super vector based methods (i.e., SDV) yield very high-dimensional data which is difficult to handle for practical applications.

4.3.1 TUCKER-3 Tensor Decomposition

To retain the spatio-temporal structure among local features, we propose to arrange SDV encoded features in the form of tensor. For this purpose, vectors $u_k$ are arranged into a $K \times M$ matrix. For $P$ different feature descriptors (i.e., MFCC, HOG, MBHx, and MBHy), the resultant $K \times M$ matrices are arranged into a rank-3 tensor. For each video segment, a super descriptor tensor of size $K \times M \times P$ is obtained.
4.3. Tensor Decomposition and Feature Selection

To discard noisy features and retain the most discriminative and compact set of features for classification, a tensor decomposition is applied. Assume a training set of $Q$ rank-3 tensors, $X_i \in \mathbb{R}^{K \times M \times P}$, $i = 1, 2, ..., Q$. The tensor decomposition of $X_i$ into three basis factors $A^{(1)} \in \mathbb{R}^{K \times J_1}$, $A^{(2)} \in \mathbb{R}^{M \times J_2}$, and $A^{(3)} \in \mathbb{R}^{P \times J_3}$, and a core feature tensor $G^i \in \mathbb{R}^{J_1 \times J_2 \times J_3}$ of total $J_1 J_2 J_3$ features, is given as

$$X_i \approx G^i \times_1 A^{(1)} \times_2 A^{(2)} \times_3 A^{(3)}, \quad (4.5)$$

where $\times_p$, $p = 1, 2, 3$, is the $p$-mode product of a tensor with a matrix. For example, let $G = [g_{j_1,j_2,j_3}]$ and $A^{(1)} = [a_{k,j_1}]$,

$$(G \times_1 A^{(1)})_{k,j_2,j_3} = \sum_{j_1} g_{j_1,j_2,j_3} a_{k,j_1}. \quad (4.6)$$

The basis factors $A^{(p)}$ can be obtained by minimizing the following cost function,

$$\min_{\{A^{(1)},A^{(2)},A^{(3)}\}} \sum_{i=1}^{Q} \|X_i - G^i \times_1 A^{(1)} \times_2 A^{(2)} \times_3 A^{(3)}\|^2_F. \quad (4.7)$$

The size of the core tensor $G^i$ is determined by calculating the number of components $J_p$ for the basis factors $A^{(p)}$. The singular values of the contracted product $X^i_{(p)} X^{i\top}_{(p)}$ are obtained,

$$X^i_{(p)} X^{i\top}_{(p)} = U \Lambda V^T. \quad (4.8)$$

Here, $\Lambda = \text{diag}(\sigma_1, \sigma_2, ..., \sigma_K)$, and $\sigma_1, \sigma_2, ..., \sigma_K$ are singular values arranged in a decreasing order. The core tensor is designed to retain the training data information equal to or above a fitness threshold $\theta$ (usually $\theta = 95\%$). Parameter $J_1$ is the smallest integer such that

$$\frac{\sum_{k=1}^{J_1} \sigma_k}{\sum_{k=1}^{K} \sigma_k} \geq \theta. \quad (4.9)$$

Parameters $J_2$ and $J_3$ are calculated in a similar manner, and thus the size of the core tensor $G^i$ is determined.

The $Q$ simultaneous standard decompositions of rank-3 tensors $X_i$ in (4.5) are equivalent to the following tensor decomposition:

$$X \approx G \times_1 A^{(1)} \times_2 A^{(2)} \times_3 A^{(3)}, \quad (4.10)$$
where the tensors $\mathbf{X} \in \mathbb{R}^{K \times M \times P \times Q}$ and $\mathbf{G} \in \mathbb{R}^{J_1 \times J_2 \times J_3 \times Q}$ are rank-4 tensors obtained by concatenating all the tensors $\mathbf{X}^i$ and $\mathbf{G}^i$ along mode-4, respectively. This decomposition model is called TUCKER-3 tensor decomposition. For the detailed mathematical model, the reader is referred to [116].

### 4.3.2 Higher-order Orthogonal Interactions

To obtain a meaningful and unique TUCKER-3 representation, orthogonality constraints are applied. For this purpose, higher-order orthogonal interactions (HOOI) algorithm [127], [128] is used, and orthogonal interactive basis factors $U^{(p)}$ are estimated for a training tensor $\mathbf{X}$. For the estimation, the factors $U^{(p)}$ are randomly initialized so that the training core tensor $\mathbf{G}$ can be obtained as

$$\mathbf{G} = \mathbf{X} \times_1 U^{(1) \, T} \times_2 U^{(2) \, T} \times_3 U^{(3) \, T}. \quad (4.11)$$

Minimizing the cost function of (4.7) is equivalent to maximizing over the matrices $U^{(p)}$ the function [127], [128],

$$J(U^{(p)}) = \| \mathbf{X} \times_1 U^{(1) \, T} \times_2 U^{(2) \, T} \times_3 U^{(3) \, T} \|_F^2. \quad (4.12)$$

If $U^{(p)}$ is fixed, the tensor $\mathbf{X}$ can be projected onto the subspace defined as

$$\mathbf{W}^{(-p)} = \mathbf{X} \times_1 U^{(1) \, T} \times_2 U^{(2) \, T} \times_3 U^{(3) \, T} = \mathbf{X} \times_{-(p,4)} \{U^T\}, \quad (4.13)$$

where $\times_{-(p,4)}$ represents the multiplication excluding mode-$p$ and mode-4. Factors $U^{(p)}$ are estimated as $I_p$ which are leading left vectors of the mode-$p$ matricized version of $\mathbf{W}^{(-p)}$. Once the basis factors $U^{(p)}$ are obtained, a test feature core tensor $\mathbf{G}^t$ for a test tensor $\mathbf{X}^t$ is obtained as $\mathbf{G}^t = \mathbf{X}^t \times \{U^T\}$.

The block diagram of the proposed tensor decomposition model is shown in Fig. 4.1. The diagram shows that the rank-3 tensors obtained after SDV encoding are concatenated first to form rank-4 tensors for the training and test datasets. Then, TUCKER-3 tensor decomposition is applied on the training tensor to get the training features and the basis factors. The basis factors are then used to extract the test features from the test tensor. Then, feature ranking is applied on the training and test features to extract salient features which are fed to a classifier in the end.
4.3. Tensor Decomposition and Feature Selection

Figure 4.1: Block diagram of the proposed SDTD method. First, the individual tensors obtained after SDV coding are concatenated to get tensors $X$ and $X^t$ for training and test datasets, respectively. The training tensor $X$ is decomposed through the TUCKER-3 tensor decomposition using the HOOI algorithm. Training features are obtained from the core tensor $G$ after the decomposition. The orthogonal basis factors $U$ are used to get the test features from $G^t = X^t \times \{U^T\}$. The feature selection is performed using Fisher ranking to select discriminative features. Finally, a classifier is used to classify the video segments.

4.3.3 Fisher Ranking for Feature Selection

The tensor model combines the local features from different descriptors to retain the spatio-temporal structure among features. Note that the higher-order discrimination techniques like tensor decomposition by themselves may not produce salient features for classification. The reason is that some discriminative features will be lost if the size of the core tensor is set too small during the tensor decomposition. However, avoiding feature loss will lead to a large core tensor and inefficient classification. Hence, there is a need for a component-wise feature selection technique to obtain salient features for classification. Fisher ranking is such a feature selection technique. The features after tensor decomposition can be sorted according to their Fisher score and the top features can be selected for classification.

In our method, the salient features for classification are selected using Fisher
4.4 Experimental Results and Analysis

The Fisher score of the qth feature, \( q = 1, 2, \ldots, J_1 J_2 J_3 \), is defined as

\[
\varphi(q) = \frac{\sum_{c=1}^{C} Q_c (\bar{g}_c^q - \bar{g}^q)^2}{\sum_{i=1}^{Q} (g^i_q - \bar{g}^q)^2},
\]

where \( g^i_q \) is the qth feature (entry) of the vectorized version of feature core tensor \( \mathbf{G}^i \), \( c \) is the class of the training sample \( \mathbf{X}^i \), \( Q_c \) is the number of training samples in class \( c \), and \( Q \) is the total number of training samples. The mean sample \( \bar{g}_c^q \) for the qth feature of the cth class and the total mean feature \( \bar{g}^q \) are defined as

\[
\bar{g}_c^q = \frac{1}{Q_c} \sum_{i \in c} g^i_q, \quad \bar{g}^q = \frac{1}{Q} \sum_{i=1}^{Q} g^i_q.
\]

The features are sorted in a descending order. The top features that achieve the best classification accuracy are selected through experimentation. The proposed SDTD model for global feature extraction is summarized in Algorithm 3 (page 60).

4.4 Experimental Results and Analysis

In this section, detailed results and analysis of the proposed SDTD model are presented. Different tensor decomposition and feature ranking techniques are evaluated and those with highest classification accuracy are chosen for the final model. The effects of model parameters are analyzed for the SDTD model, and different SDTD model configurations are evaluated. The proposed SDTD method is compared with several feature representation models in terms of classification accuracy, dimensionality, and computation time. Different classifiers are tested, and classification results of the proposed and other methods are presented for dynamic scene recognition and human interaction recognition.

4.4.1 Datasets and Evaluation Protocol

The proposed audio-visual feature representation model (i.e., SDTD) is tested for two dynamic scene recognition datasets called Maryland “in-the-wild” and YUP-PEN dynamic scenes (described in Section 3.2.5.1), and two human interaction recognition datasets called TV human interaction dataset (TVHID) [129] and Parliament [130]. The TVHID contains videos from different TV shows with dynamic
Algorithm 3 Global feature extraction via super descriptor tensor decomposition

**Input:** Feature descriptors (MFCC, HOG, MBHx, and MBHy) for each video. 

**Output:** Compact salient features $g \in \mathbb{R}_f$ for classification in each video.

1. For each descriptor, model the features with GMM using (4.1) and take the gradient of the log-likelihood of function (4.1) w.r.t mean as in (4.2).

2. Obtain the codewords $d_k \in \mathbb{R}^M$ and sparse coefficients $\alpha_{i}^{k}$ using (4.3), and make the approximations, $\mu_k = d_k$, $p_k^{+}(x_i) = \alpha_{i}^{k}$, and $\sigma_k = \sigma^2 I$ to estimate the GMM parameters in (4.2) and obtain weighted difference vectors $\alpha_{i}^{k}(x_i - d_k)$.

3. For each codeword, aggregate the weighted difference vectors through average pooling and obtain the SDV encoded features, $u_k \in \mathbb{R}^M$ using (4.4).

4. Form rank-3 tensors of size $K \times M \times P$ for $P$ descriptors (MFCC, HOG, MBHx, and MBHy) from their SDV encoded features.

5. Given a training set of tensors, decompose $X \in \mathbb{R}^{K \times M \times P \times Q}$ into core tensor $G \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times Q}$ and basis factors $U^{(1)} \in \mathbb{R}^{K \times I_1}$, $U^{(2)} \in \mathbb{R}^{M \times I_2}$, and $U^{(3)} \in \mathbb{R}^{P \times I_3}$, using (4.10), (4.11), (4.12), and (4.13).

6. Find the core tensor size by estimating the number of largest singular values using (4.8) that give $\theta$% contribution to overall singular values.

7. Obtain the test core tensor $G^t$ of the test tensor $X^t$ as $G^t = X^t \times (U^T)$.

8. Rank the test and training features in $G$ and $G^t$, respectively, using (4.14) and (4.15). Select $f$ features with the highest Fisher score.

background. There are 200 videos of four interactions: hand shake, high five, hug, and kiss. There are 100 videos of negative examples which do not contain any of the four actions. The Parliament dataset contains videos of speeches from the Greek parliament with static background. There are 228 videos of three behaviors: friendly, aggressive, and neutral. The sample video frames from the TVHID and Parliament datasets are shown in Fig. 4.2. For fair comparison, the same evaluation protocols are adopted as in [129] and [130]. That is, ten-fold and five-fold cross-validation are employed with datasets TVHID and Parliament, respectively.

### 4.4.2 Experimental Setup

This section provides details of the algorithmic implementation of the proposed SDTD method. The parameter values used in the implementation are chosen based on extensive experimental evaluation, where each parameter is varied and
4.4. Experimental Results and Analysis

(a) TVHID

(b) Parliament

Figure 4.2: Sample video frames from THIVD [129] and Parliament [130] datasets.

the value that yields the best performance is chosen.

For audio feature extraction using MFCC, the audio frame size is set to 40ms with 20ms overlap. From each frame, a 96-dimensional vector is computed, comprising MFCCs and their first and second-order derivatives. For visual feature extraction using the RDT and LRGS-STIP, the parameter settings are given in Sections 3.2.5.2 and 3.3.4.2, respectively. For Maryland and YUPPEN datasets, only visual features are extracted, whereas for TVHID and Parliament datasets both the audio and visual features are utilized.

Compared with traditional coding techniques, SDV coding can yield good results with a much smaller dictionary. Based on our experiments, a dictionary size of 500 is sufficient, beyond which there is no significant improvement in classification rate. The encoded feature vectors are arranged into a $500 \times 96$ matrix. Combining the four descriptors (MFCC, HOG, MBHx, and MBHy) yields a rank-3 tensor of size $500 \times 96 \times 4$ for each video sample. After the tensor decomposition and Fisher ranking, the number of features that gives the best classification accuracy is selected. For this purpose, different tensor decomposition algorithms and
feature ranking methods are evaluated. The TUCKER-3 tensor decomposition is implemented using the NFEA toolbox [131].

The classification performance may get biased if some of the classes in a dataset have fewer samples than the rest. During the training process, the number of samples in minority classes are increased through artificial distortions in feature space to avoid class imbalance. For this purpose, synthetic minority over-sampling technique (SMOTE) [132] is implemented for the datasets containing minority classes for better classification results.

4.4.3 Performance of Different Classifiers

The proposed recognition system is evaluated using different classifiers: \(k\)-nearest neighbor (\(k\)NN), naive Bayes (NB), non-linear support vector machines (NL-SVM), linear support vector machines (L-SVM), and extreme learning machines (ELM). The extracted audio and visual features are represented via the SDTD model and salient features that can give the best CRs are selected after Fisher ranking.

For \(k\)NN, different values of \(k\) are selected, and different distance metric are used including Euclidean, city block, cosine, and correlation distance metrics. The best results are obtained using \(k = 1\) and correlation distance metric. For NL-SVM, an RBF kernel is used for classification. For ELM classifier, a sigmoidal activation function is used with 20,000 hidden neurons. The number of neurons is selected manually for best classification results.

We compare the performance of different classifiers. The classification rates (CRs) and standard deviation (std) obtained for the proposed SDTD model are given in Table 4.1. NB classifier achieves CRs of 62.3%, 85.7%, 57.0%, and 74.6% for Maryland, YUPPEN, TVHID, and Parliament datasets, respectively. It provides the lowest CRs on all datasets in comparison with the other classifiers. \(k\)NN classifier achieves CRs of 66.9%, 92.9%, 59.3%, and 82.5% for the four datasets, and it outperforms NB classifier on all the datasets. NL-SVM achieves CRs of 85.4%, 96.9%, 73.0%, and 86.8% for Maryland, YUPPEN, TVHID, and Parliament datasets, respectively, and it outperforms NB and \(k\)NN classifiers. L-SVM provides CRs of 86.9%, 97.9%, 75.0%, and 89.0% for the four datasets. The results indicate that L-SVM performs slightly better than NL-SVM. NL-SVM works well
when the number of features is small (in the order of a few hundred). However, NL-SVM behaves like L-SVM when the number of features is large (in the order of thousands) [133]. ELM achieves CRs of 89.2%, 98.1%, 78.5%, and 93.4% for Maryland, YUPPEN, TVHID, and Parliament datasets, respectively. From the comparison of performance in Table 4.1, ELM provides the highest classification accuracy and is used for further experiments.

Table 4.1: Average CR ± std for the proposed SDTD method using naive Bayes, kNN, non-linear SVM, linear SVM, and extreme learning machines classifiers.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>NB (%)</th>
<th>kNN (%)</th>
<th>NL-SVM (%)</th>
<th>L-SVM (%)</th>
<th>ELM (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maryland</td>
<td>62.3 ± 4.3</td>
<td>66.9 ± 4.1</td>
<td>85.4 ± 3.1</td>
<td>86.9 ± 3.0</td>
<td>89.2 ± 2.7</td>
</tr>
<tr>
<td>YUPPEN</td>
<td>85.7 ± 1.7</td>
<td>92.9 ± 1.3</td>
<td>96.9 ± 0.8</td>
<td>97.9 ± 0.7</td>
<td>98.1 ± 0.7</td>
</tr>
<tr>
<td>TVHID</td>
<td>57.0 ± 2.9</td>
<td>59.3 ± 2.8</td>
<td>73.0 ± 2.6</td>
<td>75.0 ± 2.5</td>
<td>78.5 ± 2.4</td>
</tr>
<tr>
<td>Parliament</td>
<td>74.6 ± 2.9</td>
<td>82.5 ± 2.5</td>
<td>86.8 ± 2.2</td>
<td>89.0 ± 2.1</td>
<td>93.4 ± 1.6</td>
</tr>
</tbody>
</table>

4.4.4 Effects of Dictionary Size in Feature Encoding

In this section, the effects of dictionary size on CR are analyzed for the proposed method. The local feature descriptors (i.e., MFCC, HOG, MBHx, and MBHy) are encoded using SDV coding with different dictionary sizes and represented by the SDTD model. The classification results for different dictionary sizes are shown in Fig. 4.3.

For the dictionary size of 100, CRs quickly reach 64.6% for Maryland, 78.1% for YUPPEN, 63.0% for TVHID, and 72.8% for Parliament dataset. For Maryland dataset, the highest CR of 89.2% is achieved for a dictionary size of 450. For YUPPEN dataset, the highest CR of 98.1% is achieved for a dictionary size of 400. For TVHID dataset, the highest CR of 78.5% is achieved for a dictionary size of 400 codebook vectors. For Parliament dataset, the highest CR of 93.4% is achieved for a dictionary size of 450. There is no change in CRs if the dictionary sizes are increased further. The dictionary size is set to 500 for all the datasets in further experiments because there is no improvement in CRs if the dictionary size is increased further.
4.4. Experimental Results and Analysis

![Classification rate versus dictionary size for SDV coding in the SDTD model.](image)

**Figure 4.3:** Classification rate versus dictionary size for SDV coding in the SDTD model.

### 4.4.5 Evaluation of Tensor Decomposition Algorithms

Different tensor decomposition algorithms are evaluated for the proposed SDTD model. The TUCKER-3 decomposition can be obtained by applying different constraints on the basis factors in (4.7), for example, orthogonality and non-negativity [116]. Higher-order discriminant analysis (HODA) algorithm is also tested for the tensor decomposition [116]. Both HOOI and HODA algorithms are implemented in our SDTD model. The Fisher score is used to obtain the salient features for the two approaches.

The average CRs and std of both algorithms are presented in Table 4.2 for Maryland, YUPPEN, TVHID, and Parliament datasets. The HODA algorithm achieves CRs of 86.9%, 96.9%, 76.0%, and 92.1%, whereas the HOOI algorithm achieves CRs of 89.2%, 98.1%, 78.5, and 93.4%, for Maryland, YUPPEN, TVHID, and Parliament datasets, respectively. The results show that HOOI yields higher CRs than HODA on all the datasets. Therefore, HOOI algorithm is chosen in our SDTD model for tensor decomposition in further experiments.
4.4. Experimental Results and Analysis

Table 4.2: Average CR ± std for the SDTD model using HODA and HOOI algorithms.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>HODA (%)</th>
<th>HOOI (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maryland</td>
<td>86.9 ± 3.0</td>
<td>89.2 ± 2.7</td>
</tr>
<tr>
<td>YUPPEN</td>
<td>96.9 ± 0.8</td>
<td>98.1 ± 0.7</td>
</tr>
<tr>
<td>TVHID</td>
<td>76.0 ± 2.5</td>
<td>78.5 ± 2.4</td>
</tr>
<tr>
<td>Parliament</td>
<td>92.1 ± 1.8</td>
<td>93.4 ± 1.6</td>
</tr>
</tbody>
</table>

4.4.6 Effects of Fitness Threshold $\theta$ during Tensor Decomposition

In this experiment, we analyze the effects of different values of threshold $\theta$ on classification accuracy. The threshold $\theta$ is varied during the tensor decomposition which controls the core tensor size. The core tensor features after tensor decomposition are directly used for classification without any feature selection.

The average CRs using different fitness thresholds for Maryland, YUPPEN, TVHID, and Parliament datasets are given in Table 4.3. The highest CRs of 86.2%, 94.1%, 76.0%, and 89.9% are achieved using $\theta = 99\%$ for Maryland, YUPPEN, TVHID, and Parliament datasets, respectively. The CRs start decreasing when the core tensor size is reduced by selecting smaller fitness thresholds. If the threshold $\theta$ is below 90%, there is significant reduction in CRs (Table 4.3). The reason is that some discriminative features are lost if the size of the core tensor is set too small (i.e., small value of $\theta$) during the tensor decomposition. Therefore, we rely on feature selection after tensor decomposition to remove noisy and redundant features for efficient classification.

Table 4.3: Average CR ± std (in percent) for the SDTD model using different fitness thresholds $\theta$ (without feature selection).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$\theta = 99%$</th>
<th>$\theta = 90%$</th>
<th>$\theta = 70%$</th>
<th>$\theta = 50%$</th>
<th>$\theta = 30%$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maryland</td>
<td>86.2 ± 3.0</td>
<td>77.7 ± 3.7</td>
<td>72.3 ± 3.9</td>
<td>63.1 ± 4.2</td>
<td>39.2 ± 4.3</td>
</tr>
<tr>
<td>YUPPEN</td>
<td>94.1 ± 1.1</td>
<td>91.2 ± 1.3</td>
<td>85.0 ± 1.7</td>
<td>73.1 ± 2.2</td>
<td>44.8 ± 2.4</td>
</tr>
<tr>
<td>TVHID</td>
<td>76.0 ± 2.5</td>
<td>74.5 ± 2.5</td>
<td>68.5 ± 2.7</td>
<td>58.0 ± 2.8</td>
<td>34.0 ± 2.7</td>
</tr>
<tr>
<td>Parliament</td>
<td>89.9 ± 2.0</td>
<td>87.3 ± 2.2</td>
<td>79.4 ± 2.7</td>
<td>70.6 ± 3.0</td>
<td>40.4 ± 3.3</td>
</tr>
</tbody>
</table>

The fitness threshold $\theta = 99\%$ is selected for the best classification results in
further experiments. Although this value yields highest CRs, it results in a large number of features, which includes noisy and redundant features. For example, for Maryland dataset, the number of features is reduced from 144,000 (number of rank-3 tensor features) to 94,041 (number of core tensor features) for each video sample. A dimensionality reduction of 35% is achieved but 94,041 is still a very large number of features for efficient classification.

### 4.4.7 Analysis of Feature Selection after Tensor Decomposition

In this experiment, the process of feature selection after tensor decomposition is analyzed for the proposed method. After the tensor decomposition, a feature ranking method is used to rank the features in descending order of their score, and the top features are selected as input to the classifier. For this purpose, three feature ranking methods namely, Fisher ranking, Student’s t-test [134], and mutual information [135], are evaluated.

The Fisher ranking, Student’s t-test, and mutual information criteria are compared in terms of CR. The CRs as a function of the number of selected features (f) are given in Fig. 4.4 for Maryland, YUPPEN, TVHID and Parliament datasets. The highest CRs achieved by Student’s t-test criterion are 85.4% for Maryland, 96.9% for YUPPEN, 75.0% for TVHID, and 92.5% for Parliament dataset. The highest CRs achieved by mutual information method are 83.8% for Maryland, 96.9% for YUPPEN, 75.5% for TVHID, and 92.1% for Parliament dataset. The highest CRs achieved by Fisher ranking are 89.2% for Maryland, 98.1% for YUPPEN, 78.5% for TVHID, and 93.4% for Parliament dataset. In comparison with Student’s t-test and mutual information, Fisher ranking provides higher CRs. Therefore, we employ Fisher ranking for feature selection in our final SDTD model.

The feature ranking provides a significant reduction in dimensionality. For example, for Maryland dataset, the number of features is reduced from 94,041 (number of core tensor features) to 7,000 (number of features obtained after Fisher ranking) for a video (Fig. 4.4). The tensor decomposition and feature ranking discard the noisy and redundant features to obtain best classification results.
Figure 4.4: Classification rate versus number of features using Fisher ranking, Student’s t-test, and mutual information for Maryland, YUPPEN, TVHID, and Parliament datasets.

4.4.8 Analysis of Different Configurations of the SDTD Model

We evaluate three different configurations of our SDTD model. In first configuration, the audio-visual features are encoded with SDV and top features are selected using Fisher ranking (SDV+Fisher). There is no tensor decomposition performed in this case. Second, the SDV encoded features are arranged as tensors and tensor decomposition is applied to get features for classification (SDV+TD). There is no feature ranking applied in this case. Third, the complete SDTD model is used, that is, the SDV encoded features are arranged as tensors, the TUCKER-3 tensor decomposition is applied, and the top features for classification are selected through Fisher ranking (SDV+TD+Fisher).

The CRs as a function of the number of selected features ($f$) are given in Fig. 4.5 for Maryland, YUPPEN, TVHID and Parliament datasets. The SDV+TD+Fisher achieves CRs of 89.2% for Maryland, 98.1% for YUPPEN, 78.5% for TVHID, and 93.4% for Parliament dataset. These CRs are obtained by the SDV+TD+Fisher.
4.4. Experimental Results and Analysis

Figure 4.5: Classification rate versus number of features of different configurations within the SDTD model for Maryland, YUPPEN, TVHID, and Parliament datasets.

using 7,000 features for Maryland, 5,500 features for YUPPEN, 8,500 features for TVHID, and 7,000 features for Parliament dataset.

For the same numbers of features as SDV+TD+Fisher, the SDV+Fisher achieves CRs of 83.1% for Maryland, 92.4% for YUPPEN, 77.0% for TVHID, and 90.8% for Parliament dataset. If the numbers of features are increased, the CRs eventually reach the SDV+TD+Fisher configuration but for higher numbers of features, as shown in Fig. 4.5.

The SDV+TD achieves CRs of 73.1% for Maryland, 81.7% for YUPPEN, 57.0% for TVHID, and 71.1% for Parliament dataset, using the same numbers of features as the other two configurations across the four datasets. The CRs are significantly lower than the SDV+TD+Fisher and SDV+Fisher configurations for the same numbers of features.

In conclusion, the SDV+TD+Fisher configuration of our proposed SDTD model provides the best CRs for small numbers of features across the four datasets.
4.4.9 Comparison of the SDTD model with other Feature Representation Models

We compare the SDTD method with three different feature representation models: spatial pyramid matching (SPM) [58], locality-constrained linear coding (LLC) [57], and Fisher vector [61]. The SDTD method is also compared with SDV (without tensor decomposition and feature selection). In this experiment, first, we present the implementation of different feature representation models, then compare the performance of the above-mentioned models with the proposed SDTD method in terms of classification accuracy, dimensionality (number of features for classification), and computation time.

4.4.9.1 Implementation of Different Feature Representation Models

The implementation of different feature representation models is described here. For a comparison, the dictionary size is set to 500 for all the feature representation models same as in our SDTD model.

For SPM model, a dictionary is computed using \( k \)-means algorithm. Vector quantization (VQ) [55] coding is used to encode local feature descriptors. These encoded features are then pooled using max pooling and normalized using power and \( \ell_2 \) normalization. For LLC method, similar settings as SPM are used for the dictionary computation and feature pooling and normalization. For FV method, the dictionary is computed through a GMM using the expectation maximization algorithm. The encoded features are then computed through the GMM and normalized using power and \( \ell_2 \) normalization. For SDV method, encoded features are directly fed to the classifier by simply concatenating the encoded feature vectors. That is, tensor decomposition and feature selection are not performed on the encoded features. This is similar to SPM, LLC, and FV, where encoded features are concatenated to get a large single feature vector.

4.4.9.2 Comparison in terms of Classification Accuracy and Dimensionality

The performance of SPM, LLC, FV, and SDV methods is compared with the SDTD model in terms of classification accuracy and dimensionality. The total number of features \( f \) used for classification and the classification results of each method
Table 4.4: Average CR ± std of the SDTD method and four different feature representation methods: SPM, LLC, FV, and SDV.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SDTD (%)</th>
<th>SPM (%)</th>
<th>LLC (%)</th>
<th>FV (%)</th>
<th>SDV (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maryland</td>
<td>89.2 ± 2.7</td>
<td>80.0 ± 3.5</td>
<td>82.3 ± 3.4</td>
<td>86.2 ± 3.0</td>
<td>87.7 ± 2.9</td>
</tr>
<tr>
<td></td>
<td>(f=7,000)</td>
<td>(f=1,500)</td>
<td>(f=1,500)</td>
<td>(f=144,000)</td>
<td>(f=144,000)</td>
</tr>
<tr>
<td>YUPPEN</td>
<td>98.1 ± 0.7</td>
<td>94.1 ± 1.2</td>
<td>94.8 ± 1.1</td>
<td>97.4 ± 0.8</td>
<td>97.6 ± 0.7</td>
</tr>
<tr>
<td></td>
<td>(f=5,500)</td>
<td>(f=1,500)</td>
<td>(f=1,500)</td>
<td>(f=144,000)</td>
<td>(f=144,000)</td>
</tr>
<tr>
<td>TVHID</td>
<td>78.5 ± 2.4</td>
<td>71.0 ± 2.6</td>
<td>73.5 ± 2.5</td>
<td>77.0 ± 2.4</td>
<td>77.5 ± 2.4</td>
</tr>
<tr>
<td></td>
<td>(f=8,500)</td>
<td>(f=2,000)</td>
<td>(f=2,000)</td>
<td>(f=192,000)</td>
<td>(f=192,000)</td>
</tr>
<tr>
<td>Parliament</td>
<td>93.4 ± 1.6</td>
<td>85.1 ± 2.4</td>
<td>87.2 ± 2.2</td>
<td>90.8 ± 1.9</td>
<td>92.1 ± 1.8</td>
</tr>
<tr>
<td></td>
<td>(f=7,000)</td>
<td>(f=2,000)</td>
<td>(f=2,000)</td>
<td>(f=192,000)</td>
<td>(f=192,000)</td>
</tr>
</tbody>
</table>

are shown in Table 4.4.

SPM and LLC yield only 1,500 features for Maryland and YUPPEN, and 2,000 features for TVHID and Parliament datasets. SPM achieves CRs of 80.0%, 94.1%, 71.0%, and 85.1% for Maryland, YUPPEN, TVHID, and Parliament datasets, respectively. Although SPM yields a very compact representation of features, its CRs are lowest in comparison with other methods. LLC achieves CRs of 82.3%, 94.8%, 73.5%, and 87.2% for the four datasets. For the same numbers of features, LLC outperforms SPM across the four datasets.

FV and SDV yield 144,000 features for Maryland and YUPPEN, and 192,000 features for TVHID and Parliament datasets. FV achieves CRs of 86.2%, 97.4%, 77.0%, and 90.8% for Maryland, YUPPEN, TVHID, and Parliament datasets, respectively. In comparison with SPM and LLC, FV yields a large number of features for classification, and it outperforms LLC and SPM methods. SDV achieves CRs of 87.7%, 97.6%, 77.5%, and 92.1% for the four datasets. For the same numbers of features, SDV outperforms FV across the four datasets.

The SDTD discards the noisy and redundant features and yields a compact representation of features. The SDTD yields 7,000 features for Maryland, 5,500 features for YUPPEN, 8,500 features for TVHID, and 7,000 features for Parliament dataset. The SDTD method achieves CRs of 89.2%, 98.1%, 78.5%, and 93.4% for Maryland, YUPPEN, TVHID, and Parliament datasets, respectively. From comparison of the feature representation methods in Table 4.4, the proposed SDTD method achieves the highest CRs for Maryland, YUPPEN, TVHID, and
Table 4.5: Average CR ± std of the SDTD, FV, and SDV methods for the same number of features (f).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>No. of features</th>
<th>SDTD (%)</th>
<th>FV (%)</th>
<th>SDV (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maryland</td>
<td>7,000</td>
<td>89.2 ± 2.7</td>
<td>82.3 ± 3.4</td>
<td>83.1 ± 3.3</td>
</tr>
<tr>
<td>YUPPEN</td>
<td>5,500</td>
<td>98.1 ± 0.7</td>
<td>91.9 ± 1.3</td>
<td>92.4 ± 1.3</td>
</tr>
<tr>
<td>TVHID</td>
<td>8,500</td>
<td>78.5 ± 2.4</td>
<td>76.0 ± 2.5</td>
<td>77.0 ± 2.4</td>
</tr>
<tr>
<td>Parliament</td>
<td>7,000</td>
<td>93.4 ± 1.6</td>
<td>89.0 ± 2.1</td>
<td>90.7 ± 1.9</td>
</tr>
</tbody>
</table>

Parliament datasets.

The FV, SDV, and SDTD are compared in terms of classification accuracy, by selecting the same number of features using Fisher ranking. The number of features selected are 7,000 for Maryland, 5,500 for YUPPEN, 8,500 for TVHID, and 7,000 for Parliament dataset. The classification results of FV, SDV, and SDTD are given in Table 4.5. For the same numbers of features selected using Fisher score for classification, the SDTD outperforms the FV and SDV methods across the four datasets.

4.4.9.3 Comparison in terms of Computation Time

In this experiment, the proposed SDTD model is compared with other global feature representation methods: SPM, LLC, FV, and SDV, in terms of computation time. The run-time is obtained on a computer with 2.40 GHz Intel i7 CPU. The numbers of features obtained are 1,500 for SPM and LLC, 144,000 for FV and SDV, and 7,000 for SDTD method on Maryland dataset.

The computation times for different processes such as feature extraction using different methods, training of ELM classifier, and testing for classification, are given in Table 4.6. For the feature extraction, SPM takes the lowest and the SDTD method takes the highest computation time. For the training, LLC takes the lowest and FV takes the highest computation time. For the testing, LLC method takes the lowest and SDV takes the highest computation time.

Although the SDTD model outperforms SPM, LLC, FV, and SDV methods in terms of classification accuracy, the computation time of the SDTD model is a lot higher. This is because the decomposition of high dimensional tensors requires significantly more memory and computational resources. Once we obtain the final
Table 4.6: Run-time (in seconds) of different global feature representation methods: SDTD, SPM, LLC, FV, and SDV, for feature extraction, training, and testing on Maryland dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>No. of features</th>
<th>Feature extraction (s)</th>
<th>Training (s)</th>
<th>Testing (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPM</td>
<td>1,500</td>
<td>294</td>
<td>190</td>
<td>13</td>
</tr>
<tr>
<td>LLC</td>
<td>1,500</td>
<td>2,475</td>
<td>183</td>
<td>11</td>
</tr>
<tr>
<td>FV</td>
<td>144,000</td>
<td>9,400</td>
<td>4,500</td>
<td>270</td>
</tr>
<tr>
<td>SDV</td>
<td>144,000</td>
<td>26,286</td>
<td>4,140</td>
<td>285</td>
</tr>
<tr>
<td>SDTD</td>
<td>7,000</td>
<td>62,686</td>
<td>527</td>
<td>45</td>
</tr>
</tbody>
</table>

features for the classification using the proposed SDTD model, the computation time taken by the classifier to get results for a small number of features is much lower in comparison with FV and SDV methods.

4.5 Chapter Summary

In this chapter, a new method for feature representation based on tensor decomposition is presented. The local feature descriptors are first encoded using super descriptor vector coding. The encoded features are then arranged in the form of tensors to retain the spatio-temporal structure among features. To discard noisy and redundant features, the TUCKER-3 tensor decomposition is applied and discriminative features are obtained. Fisher ranking based feature selection is applied to extract salient features for classification. The proposed super descriptor tensor decomposition model is extensively evaluated in terms of classification accuracy. Different tensor decomposition and feature selection methods are evaluated and those with the best performance (i.e., HOOI algorithm and Fisher ranking) are selected for the final model.

The tensor decomposition model outperforms several other feature representation methods (i.e., SPM, LLC, FV, and SDV) in terms of classification accuracy and dimensionality. The proposed model achieves best classification rates and significant dimensionality reduction, but takes a lot more run-time in comparison with other methods. Different classifiers are tested for the proposed model; ELM classifier provides the best results.
Applications for Visual Video Recognition

Chapter contents

5.1 Introduction ....................................................... 73
5.2 Dynamic Scene Recognition ................................. 74
  5.2.1 Related Work .................................................. 74
  5.2.2 Experimental Method ......................................... 76
  5.2.3 Classification Results for Dynamic Scene Recognition ... 77
  5.2.4 Comparison with State-of-the-art Methods of Dynamic
         Scene Recognition ...................................... 79
5.3 Action Recognition ................................................ 81
  5.3.1 Related Work .................................................. 81
  5.3.2 Experimental Method ......................................... 84
  5.3.3 Classification results for Action Recognition ............. 85
  5.3.4 Comparison with State-of-the-art Methods of Action Recogn-
         ition ......................................................... 87
5.4 Chapter Summary .................................................. 90

5.1 Introduction

Visual video recognition has been widely studied in computer vision [95]–[97],
[136]–[138]. We focus here on visual video recognition for the tasks of dynamic
scene recognition and action recognition.

Automated dynamic scene recognition aims to find the overall meaning of
a dynamic scene without segmenting and recognizing individual objects in the
5.2 Dynamic Scene Recognition

5.2.1 Related Work

Over the past two decades, there have been numerous attempts to solve the dynamic scene recognition problem [95]–[97], [103], [104], [139]–[145], but state-of-the-art techniques still cannot match the human performance. The various approaches for dynamic scene recognition can be categorized as dynamic sys-
5.2. Dynamic Scene Recognition

tem models [103], [139], [140], optical flow methods [141], [142], spatio-temporal
energy filter approaches [104], [143]–[145], and deep learning methods [95]–[97].

There have been explored a few dynamic systems for modeling motion in
videos. In [139], a linear dynamic system (LDS) approach was proposed to de-
scribe dynamic texture, using a generative learning model by optimizing the
maximum likelihood. The LDS performed well for classification of dynamic tex-
tures, motion tracking, and segmentation. However, its classification accuracy
degrades for unconstrained dynamic scenes, due to its linearity assumption and
first-order Markov property. In [103], a chaotic system based method was pro-
posed to model dynamic scenes using both static and dynamic features. The
system achieved better classification accuracy than the LDS, but it is significantly
affected by intra-class variations present in dynamic scenes. In [140], slow feature
analysis (SFA) is used to develop a motion descriptor for dynamic scene recogni-
tion. The features from the SFA descriptor are integrated into a global architecture
of coding and pooling for classification. This method outperformed the LDS and
the chaotic system, but its classification accuracy is affected to some extent by
noise (i.e., camera motion, variation in scale, and view point).

Optical flow has been widely used to extract temporal features and describe
motion in videos. In [141], histogram of optical flow (HOF) descriptor was pre-
sented for modeling dynamic scenes. For this method, the classification accuracy
is negatively affected by complex motion patterns, like temporal flicker (e.g., fire
and lightening) and dynamic textures (e.g., water and multiple motions found at
a point). This is because the complex motion patterns lead to the illumination
constancy constraint being violated. In [142], a five-dimensional motion flow vec-
tor (5DMFV) is extracted from the optical flow field to get temporal information
of the scene. This method outperformed the HOF on YUPPEN dynamic scenes
dataset [104].

Spatio-temporal energy filters have been used for modeling dynamic texture
and scenes. In [143], a set of single-scale spatio-temporal energy filters is used
to model dynamic textures. The performance of these filters was improved in
[104] by extending the dynamic scene analysis to multiple scales. For multi-scale
filters, a spatio-temporal oriented energy (SOE) descriptor was presented to cap-
5.2. Dynamic Scene Recognition

ture fused static and dynamic features. While SOE outperformed the dynamic systems and optical flow methods, its classification accuracy is affected by camera motion. To deal with camera motion, the videos were processed in temporal slices in [144], and a complementary space-time orientation (CSO) descriptor was proposed based on space-time feature forests. In [145], a descriptor called bags of space-time energies (BoSE) was proposed, which uses a bank of spatio-temporal oriented filters to extract primitive features and capture the spatio-temporal structure of the video sequence. The classification accuracy was improved by BoSE using a dynamic pooling approach to deal with camera motion.

Deep learning has become very popular recently and can be naturally applied to dynamic scene recognition. In [96], a spatio-temporal feature learning using 3D convolutional neural network (CNN) was proposed for dynamic scene recognition. The learned features, called C3D (convolutional 3D), model appearance and motion information simultaneously on various video analysis tasks. In [97], statistical moments obtained using temporal aggregation of outputs of pre-trained CNNs are used as features for dynamic scene classification. This approach, referred to as SA-CNN (statistical aggregation CNN), worked well in presence of camera motion. The C3D and SA-CNN achieved the highest classification accuracy in comparison with previous methods for the dynamic scene recognition problem on YUPPEN and Maryland datasets. However, the training of deep learning architectures is time consuming.

5.2.2 Experimental Method

The proposed visual recognition system i.e., RDT+SDTD, is tested for two dynamic scene recognition datasets: Maryland “in-the-wild” and YUPPEN dynamic scenes. The descriptions of the datasets and evaluation protocol are given previously in Section 3.2.5.1. For the dynamic scene recognition task, the visual features are extracted and represented using the proposed refined dense trajectories (RDT) and super descriptor tensor decomposition (SDTD) model. The implementation details for the RDT and SDTD model on the two datasets are given previously in Sections 3.2.5.2 and 4.4.2.

The confusion matrices are obtained for the proposed method, and the relia-
5.2. Dynamic Scene Recognition

bility of the obtained CRs is measured using the Cohen’s kappa coefficient [146]. A kappa coefficient near zero means there is no significant improvement in CRs compared to just classifying the samples randomly (by chance). However, if the kappa coefficient is close to 1, then there is strong agreement between the classified labels and the ground truth, i.e., there is significant improvement over a chance classifier. To compute kappa coefficient \( \kappa \), suppose there are \( K \) categories and \( N \) total samples to be classified. The kappa coefficient is calculated as:

\[
P_\text{o} \frac{P_\text{o} - P_\text{e}}{1 - P_\text{e}}.
\]

\( P_\text{o} \) is the probability of observed agreement, that is the ratio of correctly classified samples to the total number of samples \( N \). \( P_\text{e} \) is the expected probability of chance agreement, that is 

\[
P_\text{e} = \sum_{k=1}^{K} p_{ik},
\]

where \( p_{ik} \) is the probability of the \( i \)th classifier predicting the \( k \)th class.

The statistical significance of the difference between CRs of the proposed and other methods is assessed using the Friedman’s test [105]. First, all the methods (including the proposed one) are simultaneously compared together on multiple datasets, and a \( p \)-value is obtained which indicates whether there is a significant difference between the observed classification results. If there is a significant difference, the proposed method is then compared pairwise with the other methods to see if there is a statistically significant difference between the CRs of the proposed method and those of other methods. For a significance level of 5%, if the \( p \)-value \( \leq 0.05 \), then the difference between CRs is statistically significant, otherwise it is not.

### 5.2.3 Classification Results for Dynamic Scene Recognition

The classification results of the proposed visual recognition system (RDT+SDTD) for Maryland and YUPPEN datasets are analyzed in this section. The confusion matrices for the proposed visual recognition system are given in Table 5.1. The diagonals of the two matrices give CRs of the individual classes for the two datasets. The proposed method achieves 100% accuracy for four classes (boiling water, chaotic traffic, waterfall, and waves), as shown in Table 5.1(a). The CRs of the landslide, ice-berg collapse, smooth traffic, and volcano eruption classes are reduced because of their inter-class similarities with other classes. The average CR achieved for Maryland dataset is 89.23%. In the results presented in Table 5.1(b),
### Table 5.1: Confusion matrices of the proposed visual recognition system for Maryland and YUPPEN dataset.

#### (a) Maryland

<table>
<thead>
<tr>
<th></th>
<th>Avalanche</th>
<th>Boiling Water</th>
<th>Chaotic Traffic</th>
<th>Forest Fire</th>
<th>Fountain</th>
<th>Iceberg Collapse</th>
<th>Landslide</th>
<th>Smooth Traffic</th>
<th>Tornado</th>
<th>Volcano Eruption</th>
<th>Waterfall</th>
<th>Waves</th>
<th>Whirlpool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avalanche</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Boiling Water</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Chaotic Traffic</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Forest Fire</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fountain</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Iceberg Collapse</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>80</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Landslide</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>70</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Smooth Traffic</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>80</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Tornado</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Volcano Eruption</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Waterfall</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Waves</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Whirlpool</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>90</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

#### (b) YUPPEN

<table>
<thead>
<tr>
<th></th>
<th>Beach</th>
<th>Elevator</th>
<th>Forest Fire</th>
<th>Fountain</th>
<th>Highway</th>
<th>Lightening Storm</th>
<th>Ocean</th>
<th>Railway</th>
<th>Rushing River</th>
<th>Sky-Clouds</th>
<th>Snow</th>
<th>Street</th>
<th>Waterfall</th>
<th>Windmill Farm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beach</td>
<td>96.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Elevator</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Forest Fire</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fountain</td>
<td>0</td>
<td>0</td>
<td>93.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.3</td>
<td>0</td>
<td>3.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Highway</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Lightening Storm</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Ocean</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Railway</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>96.7</td>
<td>0</td>
<td>0</td>
<td>3.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rushing River</td>
<td>3.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>96.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sky-Clouds</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Snow</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>96.7</td>
<td>0</td>
<td>3.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Street</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Waterfall</td>
<td>0</td>
<td>3.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.3</td>
<td>93.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Windmill Farm</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
there are a few inter-class similarities that affect the recognition accuracy for some classes. The average CR for YUPPEN dataset is 98.10%.

The kappa coefficients are calculated for the confusion matrices in Table 5.1. For Maryland and YUPPEN datasets, the kappa coefficients are 0.8833 and 0.9795, respectively. This shows that there is a strong agreement between the classified labels and the ground-truth.

### 5.2.4 Comparison with State-of-the-art Methods of Dynamic Scene Recognition

The proposed visual recognition system (RDT+SDTD) is compared with several state-of-the-art methods for dynamic scene recognition including HOF [141], 5DMFV [142], Chaos [103], SOE [104], SFA [140], CSO [144], BoSE [145], Imagenet [95], C3D [96], and SA-CNN [97]. The CRs of these methods are given in Table 5.2 for Maryland and YUPPEN datasets. The classification results of the above-mentioned methods are taken directly from the above references, which apply the same evaluation protocol for the two datasets. The classification results of SFA method are taken from the SFA website [147].

**Table 5.2: Average CR ± std (in percent) of the proposed RDT+SDTD and other methods for Maryland and YUPPEN datasets.**

<table>
<thead>
<tr>
<th>Method</th>
<th>Maryland</th>
<th>YUPPEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA-CNN</td>
<td>93.85 ± 2.11</td>
<td>98.33 ± 0.63</td>
</tr>
<tr>
<td>C3D</td>
<td>87.70 ± 2.88</td>
<td>98.10 ± 0.67</td>
</tr>
<tr>
<td>Imagenet</td>
<td>87.70 ± 2.88</td>
<td>96.67 ± 0.88</td>
</tr>
<tr>
<td>BoSE</td>
<td>77.69 ± 3.65</td>
<td>96.19 ± 0.93</td>
</tr>
<tr>
<td>CSO</td>
<td>67.69 ± 4.10</td>
<td>86.00 ± 1.69</td>
</tr>
<tr>
<td>SFA</td>
<td>60.00 ± 4.30</td>
<td>85.47 ± 1.72</td>
</tr>
<tr>
<td>Chaos</td>
<td>58.46 ± 4.32</td>
<td>22.86 ± 2.05</td>
</tr>
<tr>
<td>SOE</td>
<td>43.08 ± 4.34</td>
<td>80.71 ± 1.93</td>
</tr>
<tr>
<td>HOF</td>
<td>33.08 ± 4.13</td>
<td>68.33 ± 2.27</td>
</tr>
<tr>
<td>5DMFV</td>
<td>–</td>
<td>85.61 ± 1.71</td>
</tr>
<tr>
<td>RDT+SDTD</td>
<td>89.23 ± 2.72</td>
<td>98.10 ± 0.67</td>
</tr>
</tbody>
</table>

For Maryland dataset, CRs of the proposed and other methods are given in Table 5.2. The proposed method achieves a CR of 89.23%, HOF achieves the
Table 5.3: The Friedman’s test $p$-value for the proposed RDT+SDTD in comparison with other methods for Maryland dataset.

<table>
<thead>
<tr>
<th>Class</th>
<th>RDT+SDTD</th>
<th>[97]</th>
<th>[145]</th>
<th>[144]</th>
<th>[140]</th>
<th>[103]</th>
<th>[104]</th>
<th>[141]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avalanche</td>
<td>90</td>
<td>100</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td>40</td>
<td>20</td>
</tr>
<tr>
<td>Boiling Water</td>
<td>100</td>
<td>90</td>
<td>70</td>
<td>80</td>
<td>70</td>
<td>60</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Chaotic Traffic</td>
<td>100</td>
<td>100</td>
<td>90</td>
<td>90</td>
<td>80</td>
<td>70</td>
<td>60</td>
<td>30</td>
</tr>
<tr>
<td>Forest Fire</td>
<td>90</td>
<td>100</td>
<td>90</td>
<td>80</td>
<td>10</td>
<td>60</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>Fountain</td>
<td>90</td>
<td>90</td>
<td>70</td>
<td>80</td>
<td>50</td>
<td>60</td>
<td>50</td>
<td>20</td>
</tr>
<tr>
<td>Iceberg Collapse</td>
<td>80</td>
<td>100</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>20</td>
</tr>
<tr>
<td>Landslide</td>
<td>70</td>
<td>90</td>
<td>60</td>
<td>30</td>
<td>60</td>
<td>30</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Smooth Traffic</td>
<td>80</td>
<td>90</td>
<td>70</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Tornado</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>80</td>
<td>70</td>
<td>80</td>
<td>70</td>
<td>40</td>
</tr>
<tr>
<td>Volcano Eruption</td>
<td>80</td>
<td>90</td>
<td>80</td>
<td>70</td>
<td>80</td>
<td>70</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>Waterfall</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>50</td>
<td>50</td>
<td>40</td>
<td>60</td>
<td>20</td>
</tr>
<tr>
<td>Waves</td>
<td>100</td>
<td>90</td>
<td>90</td>
<td>80</td>
<td>60</td>
<td>80</td>
<td>50</td>
<td>80</td>
</tr>
<tr>
<td>Whirlpool</td>
<td>90</td>
<td>90</td>
<td>80</td>
<td>70</td>
<td>80</td>
<td>50</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>$p$-value</td>
<td>RDT+SDTD vs.</td>
<td>1.57x10^{-1}</td>
<td>2.70x10^{-3}</td>
<td>3.11x10^{-4}</td>
<td>5.32x10^{-4}</td>
<td>3.11x10^{-4}</td>
<td>3.11x10^{-4}</td>
<td>3.11x10^{-4}</td>
</tr>
</tbody>
</table>

The lowest CR of 33.08%, and SA-CNN achieves the highest CR of 93.85%. In terms of CR, the proposed method outperforms all other methods except SA-CNN. The $p$-values from the comparison of the proposed RDT+SDTD versus other methods are given in the last row of Table 5.3. The RDT+SDTD is compared (using the Friedman’s test) to only those methods which provide class-wise CRs for the dataset. A $p$-value of $8.16x10^{-14}$ (significantly lower than 0.05) is obtained when comparing all the methods in Table 5.3 simultaneously. This shows that the methods have significantly different performances. The RDT+SDTD achieves a statistically significant improvement over most of the methods (BoSE, CSO, SFA, SOE, HOF, and Chaos), as the $p$-values are less than 0.05. Although the differences between CRs of RDT+SDTD and the deep learning methods: C3D and Imagenet, are not significant, the RDT+SDTD still outperforms these two methods. Only SA-CNN achieves a higher CR than the RDT+SDTD but the difference is not statistically significant ($p$-value = 0.16, greater than 0.05, in comparison with SA-CNN).

For YUPPEN dataset, CRs of the proposed and other methods are given in Table 5.2. The proposed RDT+SDTD method achieves a CR of 98.10%, Chaos achieves the lowest CR of 22.86%, and SA-CNN achieves the highest CR of 98.33%. From the results, the proposed method achieves a higher CR than all other methods except SA-CNN. The $p$-values from the comparison of the RDT+SDTD versus other methods are given in Table 5.4. The RDT+SDTD is compared (using the
### 5.3. Action Recognition

#### 5.3.1 Related Work

Human activity recognition has been an active research area in computer vision for over a few decades. Detailed reviews on action recognition methods and tasks are provided in [137], [138], and [148], we therefore mention here a few recent methods. The different methods for action recognition can be categorized as space-time methods [23], [28], [112], [149]–[151], appearance based methods [30], [152]–[154], and learning based methods (including deep learning) [155]–[159].

Many approaches for human activity recognition extract space-time features

---

Table 5.4: The Friedman’s test $p$-value for the proposed RDT+SDTD in comparison with other methods for YUPPEN dataset.

<table>
<thead>
<tr>
<th>Class</th>
<th>RDT+SDTD</th>
<th>[97]</th>
<th>[145]</th>
<th>[144]</th>
<th>[142]</th>
<th>[146]</th>
<th>[104]</th>
<th>[141]</th>
<th>[103]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beach</td>
<td>97</td>
<td>97</td>
<td>100</td>
<td>100</td>
<td>97</td>
<td>93</td>
<td>93</td>
<td>87</td>
<td>30</td>
</tr>
<tr>
<td>Elevator</td>
<td>100</td>
<td>100</td>
<td>97</td>
<td>100</td>
<td>90</td>
<td>97</td>
<td>100</td>
<td>87</td>
<td>47</td>
</tr>
<tr>
<td>Forest Fire</td>
<td>100</td>
<td>100</td>
<td>93</td>
<td>83</td>
<td>80</td>
<td>70</td>
<td>67</td>
<td>63</td>
<td>17</td>
</tr>
<tr>
<td>Fountain</td>
<td>93</td>
<td>100</td>
<td>87</td>
<td>47</td>
<td>60</td>
<td>57</td>
<td>43</td>
<td>43</td>
<td>3</td>
</tr>
<tr>
<td>Highway</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>73</td>
<td>87</td>
<td>93</td>
<td>70</td>
<td>47</td>
<td>23</td>
</tr>
<tr>
<td>Lightning Storm</td>
<td>100</td>
<td>93</td>
<td>97</td>
<td>93</td>
<td>67</td>
<td>87</td>
<td>77</td>
<td>63</td>
<td>37</td>
</tr>
<tr>
<td>Ocean</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>90</td>
<td>90</td>
<td>100</td>
<td>100</td>
<td>97</td>
<td>43</td>
</tr>
<tr>
<td>Railway</td>
<td>97</td>
<td>100</td>
<td>100</td>
<td>93</td>
<td>87</td>
<td>93</td>
<td>80</td>
<td>83</td>
<td>7</td>
</tr>
<tr>
<td>Rushing River</td>
<td>97</td>
<td>100</td>
<td>97</td>
<td>97</td>
<td>93</td>
<td>87</td>
<td>93</td>
<td>77</td>
<td>10</td>
</tr>
<tr>
<td>Sky-Clouds</td>
<td>100</td>
<td>100</td>
<td>97</td>
<td>100</td>
<td>93</td>
<td>93</td>
<td>83</td>
<td>87</td>
<td>47</td>
</tr>
<tr>
<td>Snow</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>57</td>
<td>90</td>
<td>70</td>
<td>87</td>
<td>47</td>
<td>10</td>
</tr>
<tr>
<td>Street</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>90</td>
<td>77</td>
<td>17</td>
</tr>
<tr>
<td>Waterfall</td>
<td>93</td>
<td>97</td>
<td>83</td>
<td>77</td>
<td>77</td>
<td>73</td>
<td>63</td>
<td>47</td>
<td>10</td>
</tr>
<tr>
<td>Windmill Farm</td>
<td>100</td>
<td>93</td>
<td>100</td>
<td>93</td>
<td>93</td>
<td>87</td>
<td>83</td>
<td>53</td>
<td>17</td>
</tr>
</tbody>
</table>

$p$-value RDT+SDTD vs. 1.57×10⁻¹ 5.27×10⁻¹ 2.09×10⁻¹ 1.30×10⁻¹ 3.11×10⁻¹ 5.32×10⁻¹ 1.82×10⁻¹ 1.82×10⁻¹

Friedman’s test) to only those methods which provide CRs of individual classes for the dataset. A $p$-value of 7.52×10⁻¹⁶ (significantly lower than 0.05) is obtained when all the methods in Table 5.4 are compared simultaneously. This means that the methods achieve significantly different CRs. The RDT+SDTD achieves a statistically significant improvement over most of the methods (CSO, 5DMFV, SFA, SOE, HOF, and Chaos), as the $p$-values are less than 0.05. The RDT+SDTD outperforms Imagenet and BoSE methods. The RDT+SDTD achieves the same CR as C3D. Only SA-CNN achieves a higher CR than the RDT+SDTD but the difference is not statistically significant, that is $p$-value = 0.16, greater than 0.05, in comparison with SA-CNN.
using motion trajectories for modeling motion dynamics. For example, in [23], a dense trajectories and motion boundary descriptors based video representation was introduced for modeling motion. Trajectories are formed by tracking densely sampled interest points, then multiple appearance and motion descriptors are computed along the trajectories. This method consistently performed well on many small and large-scale action recognition datasets. In [28], a selective STIP detector based on surround suppression and temporal constraints was introduced. The motion trajectories are formed by tracking the STIPs, and N-jet features are extracted and represented by BoW model for action recognition. In addition, an actor specific spatio-temporal clustering of STIPs was presented for automatic annotation, and classification accuracy was improved. In [65], a stacked Fisher vector method to represent spatio-temporal features extracted from densely sampled large sub-volumes of videos was proposed. The feature extraction process was based on dense trajectories method in [23]. The stacked Fisher vector approach outperformed many state-of-the-art methods for action recognition.

There are some space-time feature extraction approaches which use alternative strategies other than the motion trajectories. For example, in [112], a framework for recognizing human actions from videos “in-the-wild” was proposed using static and motion features. The hybrid features are later pruned using motion cues, and PageRank is used to extract most informative static features. The use of hybrid static and motion features proved to be effective for recognizing human actions. In [149], action recognition is addressed by characterizing actions as 3D objects using probability distributions of spatio-temporal features. The Lie-algebraized Gaussian method is employed for mid-level feature representation. The classification accuracy was improved on different datasets in comparison with previous methods. In [150], semantically rich features are extracted by pooling of a large number of smaller action detectors. A high-level representation is built using a large bank of individual and viewpoint tuned action detectors. This method achieved state-of-the-art results on different action recognition datasets. In [151], a region-based mixture model was proposed to classify human actions. A set of long-term motion trajectories and common shape is extracted from the
low-resolution videos to obtain a dense representation. The spatial layout of the features is encoded through the mixture model for classification. This approach works well for the classification of actions in low-resolution videos.

Appearance based methods focus on capturing shape and geometry of the human actions. In [152], human action recognition is performed using 3D reconstruction data. The spatial information, global motion, and 3D shape for action recognition are described by local-level 3D flow, global-level 3D flow, and global-temporal shape descriptor, respectively. This approach improved the classification accuracy in comparison with previous methods but it is computationally expensive. In [30], a global feature descriptor was proposed to capture geometrical distribution of STIPs using an extended 3D discrete Radon transform. The descriptor was designed to be robust to noise and invariant to geometric transformation. In addition, a fusion technique was presented to combine Radon features with the BoW model. In [153], a method that extracts negative-space based features from the surroundings of subjects was introduced. These features tend to be robust to occlusion, small shadows, and deformed actions. In [154], a framework was presented that integrates a spatial distribution of edge gradient of poses and geometric orientation of a human silhouette in videos. This method extracts features rich of appearance and angular kinematics information, which provide discriminative depiction for action recognition by significantly wrapping the local and global information.

Learning based methods including deep learning have become very famous recently and can naturally be applied for action recognition. In [155], a subspace learning framework based on kernelized multi-view projection was proposed for action recognition. The method semantically embeds a variety of features to achieve dimensionality reduction for multi-view data. It improved the classification accuracy on different action recognition datasets by combining multiple features. In [156], a learning based method was introduced for action recognition based on sparse coding techniques. A set of representative atomic actions acts is obtained by decomposing videos to handle intra-class variations. An inter-temporal relational act descriptor was also presented to capture relative similarity relations between the atomic acts. In [157], an unsupervised feature learning tech-
technique based on independent subspace analysis algorithm was proposed to learn spatio-temporal features from unlabeled videos. This method was shown to work well for learning hierarchical representations when combined with deep learning like stacking and convolution. In [158], a continuous human activity learning framework was presented for streaming videos using deep networks and active learning. This framework selects features from unlabeled incoming videos and incrementally improves the model. In [159], a method that combines slow feature analysis with deep learning was introduced. The structural features from the videos are captured by a learning structure of a two-layered slow feature analysis with 3D convolution and max pooling. The slow feature analysis has also been used in [140].

The performance of the above-mentioned space-time feature extraction methods depends on efficient feature extraction. These methods can be tailored for specific tasks. Although space-time feature extraction methods model motion effectively, their performance is affected due to noise like camera motion. On the other hand, the shape and appearance based methods are not affected much by camera motion but they inherit model complexity. The learning and deep learning methods achieve higher classification accuracy than the space-time feature extraction and appearance based methods. Although the deep learning methods perform well, the training of deep learning architectures is time consuming.

5.3.2 Experimental Method

The proposed visual recognition system is tested for the task of action recognition using KTH, UCF, and YouTube datasets. The descriptions of these datasets and evaluation protocol used are given previously in Section 3.3.4.1. For the task of action recognition, the visual features are extracted using LRGS-STIP detector and represented by the SDTD model. The implementation details of the LRGS-STIP detector and the SDTD model are given previously in Sections 3.3.4.2 and 4.4.2, respectively.

For the above-mentioned datasets, the confusion matrices are obtained for the proposed method, and the reliability of the obtained CRs is measured using the Cohen’s kappa coefficient [146]. For a comparison of the proposed visual recog-
5.3. Action Recognition

In order to compare the performance of the proposed action recognition system with other methods, the Friedman’s test [105] is used to assess the statistical significance of the differences between CRs of the proposed and other methods. The descriptions of the Cohen’s kappa coefficient and the Friedman’s test are given in Section 5.2.2.

5.3.3 Classification results for Action Recognition

The classification results for the proposed visual recognition system (LRGS-STIP+SDTD) are analyzed in this section. The classification results are obtained using the best configuration of our SDTD model, as concluded in Chapter 4. For this purpose, SDV is used for feature encoding (with dictionary size of 500), HOOI algorithm is used for tensor decomposition, Fisher ranking is used for feature selection, and ELM is used for classification. The CRs as a function of number of features are given in Fig. 5.1 for action recognition datasets: KTH, UCF, and YouTube. The highest CR of 97.5% is achieved for KTH dataset using 5,500 features. The highest CR of 93.3% is achieved for UCF dataset using 6,500 features. The highest CR of 91.5% is achieved for YouTube dataset using 7,500 features.

![Figure 5.1: Classification rate versus number of features for KTH, UCF, and YouTube datasets.](image)

Figure 5.1: Classification rate versus number of features for KTH, UCF, and YouTube datasets.

In the next experiment, the classification results for individual classes are analyzed for the action recognition datasets. The confusion matrices of the clas-
Table 5.5: Confusion matrices of the proposed visual recognition system for action recognition using KTH, UCF, and YouTube datasets.

(a) KTH

<table>
<thead>
<tr>
<th></th>
<th>Walking</th>
<th>Jogging</th>
<th>Running</th>
<th>Boxing</th>
<th>Clapping</th>
<th>Waving</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking</td>
<td>97</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Jogging</td>
<td>2</td>
<td>97</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Running</td>
<td>0</td>
<td>1</td>
<td>99</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Boxing</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>97</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Clapping</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>98</td>
<td>2</td>
</tr>
<tr>
<td>Waving</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>97</td>
<td></td>
</tr>
</tbody>
</table>

(b) UCF

<table>
<thead>
<tr>
<th></th>
<th>Diving</th>
<th>Golf swing</th>
<th>Kicking</th>
<th>Lifting</th>
<th>Riding horse</th>
<th>Running</th>
<th>Skate boarding</th>
<th>Swing-bench</th>
<th>Swing-side</th>
<th>Walking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diving</td>
<td>92.9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>71</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Golf swing</td>
<td>0</td>
<td>88.9</td>
<td>0</td>
<td>8.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5.5</td>
<td>0</td>
</tr>
<tr>
<td>Kicking</td>
<td>0</td>
<td>5</td>
<td>95</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Lifting</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Riding horse</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>91.7</td>
<td>8.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Running</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>92.3</td>
<td>7.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Skate boarding</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Swing-bench</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>95</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Swing-side</td>
<td>0</td>
<td>7.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>92.3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Walking</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>90.9</td>
<td>0</td>
</tr>
</tbody>
</table>

(c) YouTube

<table>
<thead>
<tr>
<th></th>
<th>Basketball shooting</th>
<th>Volleyball spiking</th>
<th>Trampoline jumping</th>
<th>Soccer juggling</th>
<th>Horseback riding</th>
<th>Cycling</th>
<th>Diving</th>
<th>Golf swinging</th>
<th>Tennis swinging</th>
<th>Walking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basketball shooting</td>
<td>86</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Volleyball spiking</td>
<td>5</td>
<td>87</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Trampoline jumping</td>
<td>3</td>
<td>1</td>
<td>94</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Soccer juggling</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>91</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Horseback riding</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>86</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Cycling</td>
<td>0</td>
<td>0</td>
<td>2.1</td>
<td>2.8</td>
<td>95.2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Diving</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>94</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Swinging</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>93</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Golf swinging</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>92</td>
<td>3</td>
</tr>
<tr>
<td>Tennis swinging</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>90</td>
<td>2</td>
</tr>
<tr>
<td>Walking</td>
<td>1.6</td>
<td>0</td>
<td>0.8</td>
<td>1.6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>95.9</td>
</tr>
</tbody>
</table>
5.3. Action Recognition

Table 5.6: Average CR ± std (in percent) of the proposed LRGS-STIP+SDTD and other methods for KTH, UCF, and YouTube datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>KTH</th>
<th>UCF</th>
<th>YouTube</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sadanand et al. [150]</td>
<td>98.2 ± 0.5</td>
<td>95.0 ± 1.8</td>
<td>–</td>
</tr>
<tr>
<td>Alfaro et al. [156]</td>
<td>97.5 ± 0.6</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Chen et al. [149]</td>
<td>97.4 ± 0.6</td>
<td>92.7 ± 2.1</td>
<td>–</td>
</tr>
<tr>
<td>Hasan et al. [158]</td>
<td>96.6 ± 0.7</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Chakraborty et al. [28]</td>
<td>96.4 ± 0.8</td>
<td>–</td>
<td>87.0 ± 1.0</td>
</tr>
<tr>
<td>Vishwakarma et al. [154]</td>
<td>95.5 ± 0.8</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Yuan et al. [30]</td>
<td>95.5 ± 0.8</td>
<td>87.3 ± 2.7</td>
<td>–</td>
</tr>
<tr>
<td>Wang et al. [23]</td>
<td>95.3 ± 0.9</td>
<td>89.1 ± 2.5</td>
<td>85.4 ± 1.0</td>
</tr>
<tr>
<td>Rahman et al. [153]</td>
<td>94.5 ± 0.9</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Le et al. [157]</td>
<td>93.9 ± 1.0</td>
<td>86.5 ± 2.8</td>
<td>75.8 ± 1.3</td>
</tr>
<tr>
<td>Liu et al. [112]</td>
<td>93.8 ± 1.0</td>
<td>–</td>
<td>71.2 ± 1.3</td>
</tr>
<tr>
<td>Sun et al. [159]</td>
<td>93.1 ± 1.0</td>
<td>86.6 ± 2.8</td>
<td>–</td>
</tr>
<tr>
<td>Peng et al. [65]</td>
<td>–</td>
<td>–</td>
<td>93.4 ± 0.7</td>
</tr>
<tr>
<td>LRGS-STIP+SDTD</td>
<td>97.5 ± 0.6</td>
<td>93.3 ± 2.0</td>
<td>91.5 ± 0.8</td>
</tr>
</tbody>
</table>

Classification results are given in Table 5.5 for KTH, UCF, and YouTube datasets. For KTH dataset, CRs higher than 97% are achieved for most of the classes. For UCF dataset, CRs higher than 90% are achieved for nine out of ten classes, where CRs of 100% are achieved for lifting and skate boarding classes. The classes namely walking, golf swing, and riding horse are affected by inter-class similarities with other classes. For YouTube dataset, CRs higher than 90% are achieved for eight out of eleven classes. There are significant inter-class similarities which affect the CRs of the other classes, especially horseback riding, volleyball spiking, and basketball shooting.

The kappa coefficients are calculated for the confusion matrices in Table 5.5. For KTH, UCF, and YouTube datasets, the kappa coefficients are 0.9700, 0.9322, and 0.9040, respectively. This shows that there is a strong agreement between the classified labels and the ground-truth labels.

5.3.4 Comparison with State-of-the-art Methods of Action Recognition

The proposed visual recognition system (LRGS-STIP+SDTD) is compared with some state-of-the-art methods: space-time based [23], [28], [112], [149]–[150],

87
5.3. Action Recognition

[157], appearance based [30], [153], [154], and learning based methods (including deep learning) [156], [158], [159]. The CRs of the proposed approach and other methods are given in Table 5.6. The CRs of different methods are directly taken from the above references, which apply the same evaluation protocol.

For KTH dataset, the proposed method achieves a CR of 97.5%, and it outperforms most of the existing methods, only method in [150] (called action bank) has a slightly better CR than the proposed method. The p-values from the comparison of the proposed LRGS-STIP+SDTD versus other methods are given in the last row of Table 5.7. The LRGS-STIP+SDTD is compared (using the Friedman’s test) to only those methods which provide CRs of every class for the dataset. A p-value of $7.39 \times 10^{-2}$ (greater than 0.05) is obtained when comparing all the methods in Table 5.7 simultaneously. Also, the p-values are greater than 0.05 when the LRGS-STIP+SDTD is compared with all other methods individually (Table 5.7). This shows that all the methods (including the proposed one) have almost similar performance, and there is no significant difference between their CRs. This is because, KTH dataset contains stable and homogenous backgrounds with simple actions performed by a single actor. Although the CR of action bank method is slightly higher than our method, the difference between the CRs is not statistically significant ($p$-value = 0.18, greater than 0.05).

Table 5.7: The Friedman’s test p-value for the proposed LRGS-STIP+SDTD in comparison with other methods for KTH dataset.

<table>
<thead>
<tr>
<th>Class</th>
<th>LRGS-STIP+SDTD</th>
<th>[150]</th>
<th>[149]</th>
<th>[158]</th>
<th>[30]</th>
<th>[153]</th>
<th>[112]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking</td>
<td>97</td>
<td>97</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>98</td>
<td>96</td>
</tr>
<tr>
<td>Jogging</td>
<td>97</td>
<td>100</td>
<td>95</td>
<td>93</td>
<td>87</td>
<td>95</td>
<td>83</td>
</tr>
<tr>
<td>Running</td>
<td>99</td>
<td>100</td>
<td>90</td>
<td>93</td>
<td>90</td>
<td>91</td>
<td>84</td>
</tr>
<tr>
<td>Boxing</td>
<td>97</td>
<td>92</td>
<td>100</td>
<td>94</td>
<td>100</td>
<td>90</td>
<td>99</td>
</tr>
<tr>
<td>Clapping</td>
<td>98</td>
<td>100</td>
<td>99</td>
<td>100</td>
<td>99</td>
<td>95</td>
<td>94</td>
</tr>
<tr>
<td>Waving</td>
<td>97</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>97</td>
<td>98</td>
<td>95</td>
</tr>
<tr>
<td>p-value</td>
<td>LRGS-STIP+SDTD vs.</td>
<td>0.18</td>
<td>0.41</td>
<td>1.00</td>
<td>0.66</td>
<td>0.41</td>
<td>0.18</td>
</tr>
</tbody>
</table>

For UCF dataset, the proposed method achieves a CR of 93.3%, and it outperforms all other methods except action bank method [150]. The action bank provides a high-level representation of videos based on a large number of action
detectors. The high-level features are shown to be superior than the low-level features for discriminating videos [150]. The LRGS-STIP+SDTD is compared (using the Friedman’s test) to only those methods which provide CRs of individual categories for the dataset. The $p$-values from the comparison of the proposed versus other methods are given in Table 5.8. A $p$-value of 0.55 is obtained when comparing all the methods in Table 5.8 simultaneously. The $p$-values from individual comparison of the LRGS-STIP+SDTD with other methods are also greater than 0.05. This means that the methods in Table 5.8 have no significant difference between their CRs. Although the CR of action bank method is marginally higher than the proposed method, there is no statistically significant difference between their CRs ($p$-value = 1.00, greater than 0.05).

Table 5.8: The Friedman’s test $p$-value for the proposed LRGS-STIP+SDTD in comparison with other methods for UCF dataset.

<table>
<thead>
<tr>
<th>Class</th>
<th>LRGS-STIP+SDTD</th>
<th>[150]</th>
<th>[149]</th>
<th>[30]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diving</td>
<td>93</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Golf swing</td>
<td>89</td>
<td>100</td>
<td>100</td>
<td>89</td>
</tr>
<tr>
<td>Kicking</td>
<td>95</td>
<td>100</td>
<td>85</td>
<td>95</td>
</tr>
<tr>
<td>Lifting</td>
<td>100</td>
<td>83</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Riding horse</td>
<td>92</td>
<td>100</td>
<td>83</td>
<td>58</td>
</tr>
<tr>
<td>Running</td>
<td>92</td>
<td>91</td>
<td>69</td>
<td>69</td>
</tr>
<tr>
<td>Skate boarding</td>
<td>100</td>
<td>92</td>
<td>83</td>
<td>83</td>
</tr>
<tr>
<td>Swing-bench</td>
<td>95</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Swing-side</td>
<td>92</td>
<td>89</td>
<td>100</td>
<td>77</td>
</tr>
<tr>
<td>Walking</td>
<td>91</td>
<td>86</td>
<td>100</td>
<td>91</td>
</tr>
</tbody>
</table>

$p$-value LRGS-STIP+SDTD vs. 1.00 0.74 0.41

For YouTube dataset, the proposed visual recognition system achieves a CR of 91.5% which is second highest among other methods. The method in [65] has a higher CR than our method because it uses mid-level feature representation by multiple Fisher encoding in a hierarchical architecture. The proposed method is compared (using the Friedman’s test) only to the method in [112], because no other method provides the CRs of individual classes. A $p$-value of $9.11 \times 10^{-4}$ (significantly lower than 0.05) is obtained when comparing the LRGS-STIP+SDTD...
with the method in [112]. This shows that two methods have significantly different performance.

5.4 Chapter Summary

In this chapter, we analyze the performance of our proposed visual recognition system for the tasks of dynamic scene recognition and action recognition. To extract visual features, the proposed RDT and LRGS-STIP methods are used for dynamic scene recognition and action recognition tasks, respectively. The extracted visual features are then represented through the SDTD model for classification. The classification results of the proposed visual recognition system are compared with ground-truth and other methods using the Cohen’s kappa coefficient and the Friedman’s test. The proposed recognition system achieves 100% classification rates for many classes of Maryland, YUPPEN, and UCF datasets. The results show that the proposed visual recognition system outperforms many state-of-the-art methods for the tasks of dynamic scene recognition and action recognition.
Exploiting audio-visual information can greatly improve the performance of video recognition systems. Such audio-visual recognition systems have been proposed
for the tasks of human interaction recognition and violent scene detection. Human interaction recognition deals with understanding of behaviors and interactions between people (e.g., high five and hand shake) in complex scenes. Recognizing human behaviors can provide important information about the psychological state and personality of a person [10]. The applications of human interaction and behavior recognition relate to automatic surveillance, video indexing, video retrieval, and human-computer interaction. Human interaction recognition in videos is a challenging problem. The different challenges include large intra-class variabilities and lighting conditions. In addition, the time duration of human interactions is short in videos which makes the problem difficult.

The video material including television programs, movies, and internet videos has increased rapidly in the last few decades. The ease of accessibility to a huge video enterprise via video-on-demand has raised the necessity of filtering the video content. The applications range from surveillance to parental control. For example, violence can affect a child’s personality in a harmful way and it is important for parents to filter such content. Although there are different movie ratings available, the interpretation of the word violence varies from one individual to another. The material uploaded online usually does not have any content description in terms of violence. With this in view, there is a need to develop some methods to analyze the video content.

In this chapter, we apply our proposed audio-visual recognition system for the tasks of human interaction recognition and violent scene detection. Firstly, the related work in the fields of human interaction recognition and violent scene detection is discussed in Sections 6.2.1 and 6.3.1, respectively. Secondly, the classification results are presented for the two applications in Sections 6.2.3 and 6.3.3. In the end, the proposed audio-visual recognition system is compared with some of the state-of-the-art methods for the above tasks in Sections 6.2.4 and 6.3.4.

6.2 Human Interaction Recognition

6.2.1 Related Work

The various approaches for human interaction recognition can be categorized based on different types of features they extract: visual features based [129], [130],
6.2. Human Interaction Recognition

[160], [161] and audio-visual based methods [1], [2].

Many approaches extract only visual features for modelling human interactions. For example, in [160], a time-series based method for human interaction recognition was presented by exploiting the temporal information in video data. The Hankel matrices are built to provide dynamic information of the data. The method shows that the principal angles between the subspaces that represent different human interactions, can be used to compare the interactions in videos. In [129], a person-centric approach for human interaction recognition was proposed using head and body trackers. In this method, upper bodies and heads are tracked along with occlusion detection to obtain robust person tracks. A descriptor was presented to extract the spatio-temporal information around different head orientations. In [130], a conditional random fields based method was presented for human behavior recognition. This method employs multiple spatio-temporal features like STIP and HOG3D, and kinematics features such as velocity and acceleration of the subjects. The human behaviors like friendly, aggressive, and neutral are classified for different speakers of Greek parliament. In [161], a Hough voting extension was proposed that can provide fast and efficient interest point matching for human interaction recognition. A feature matching is performed using random projection trees to leverage data distribution of local features. Above visual feature extraction based methods for human interaction recognition can be ambiguous to machines because of inter-class similarities, if only visual information is taken into account.

Some approaches combine the visual and auditory cues for better classification accuracy. For example, in [1], both audio and visual features are extracted and represented by an audio-visual bag-of-words model for human interaction recognition. The audio features are extracted using Mel-frequency cepstral coefficients (MFCC) and visual features are obtained from multiple descriptors computed within volumes around detected STIPs. It was shown that the combined audio and visual information yields better classification accuracy than visual only information. In [2], a hidden conditional random field method using multi-modal features was presented for human behavior recognition. A supervised framework is built for modelling of individual (e.g., high five and hand shake) and social behav-
6.2. Human Interaction Recognition

iors (e.g., friendly and aggressive). The audio features are obtained using MFCC and visual features are extracted using STIPs, head orientation, and proxemic features. Prior to the feature fusion, canonical correlation analysis is employed to find correlation between visual and auditory features. This method outperformed many previous methods for human interaction and behavior recognition.

6.2.2 Experimental Method

The proposed audio-visual recognition system (LRGS-STIP+SDTD) is tested for two human interaction recognition datasets: TVHID and Parliament. The descriptions of the datasets and evaluation protocol are given previously in Section 4.4.1. The audio features are extracted using MFCC and visual features are extracted using the LRGS-STIP method. The bi-modal features are then represented by the SDTD model for classification. The implementation details for MFCC, the LRGS-STIP, and the SDTD are described previously in Sections 4.4.2 and 3.3.4.2.

The confusion matrices are obtained for the proposed audio-visual recognition system and the reliability of the obtained CRs is measured using the Cohen’s kappa coefficient [146]. The statistical significance of the difference between CRs of the proposed and other methods is assessed using the Friedman’s test [105]. The Cohen’s kappa coefficient and the Friedman’s test are described previously in Section 5.2.2.

6.2.3 Classification Results for Human Interaction Recognition

The proposed audio-visual recognition system i.e., LRGS-STIP+SDTD, is tested for the task of human interaction recognition on TVHID and Parliament datasets. For this purpose, audio-visual features are extracted using MFCC and the LRGS-STIP, and represented by the SDTD model. The confusion matrices of the classification results are given in Table 6.1 for TVHID and Parliament datasets.

For TVHID dataset, a CR of 84% is achieved for class high five, and CRs of more than 76% are achieved for three out of four classes. There are observed inter-class similarities which affect the CRs of different classes. For example, class hand shake is misidentified as high five and kiss, and class hug is misidentified as kiss. For Parliament dataset, a CR of 95.9% is achieved for class aggressive.
6.2. Human Interaction Recognition

Table 6.1: Confusion matrices of the proposed audio-visual recognition system for human interaction recognition datasets: TVHID and Parliament.

(a) TVHID

<table>
<thead>
<tr>
<th></th>
<th>Hand shake</th>
<th>High five</th>
<th>Hug</th>
<th>Kiss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand shake</td>
<td>76</td>
<td>10</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>High five</td>
<td>6</td>
<td>84</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Hug</td>
<td>6</td>
<td>2</td>
<td>78</td>
<td>14</td>
</tr>
<tr>
<td>Kiss</td>
<td>6</td>
<td>4</td>
<td>14</td>
<td>76</td>
</tr>
</tbody>
</table>

(b) Parliament

<table>
<thead>
<tr>
<th></th>
<th>Friendly</th>
<th>Aggressive</th>
<th>Neutral</th>
</tr>
</thead>
<tbody>
<tr>
<td>Friendly</td>
<td>93.3</td>
<td>1.1</td>
<td>5.5</td>
</tr>
<tr>
<td>Aggressive</td>
<td>1.4</td>
<td>95.9</td>
<td>2.7</td>
</tr>
<tr>
<td>Neutral</td>
<td>6.2</td>
<td>3.1</td>
<td>90.8</td>
</tr>
</tbody>
</table>

All the classes in Parliament dataset contain speakers giving speeches, there are significant inter-class similarities in visual features. In this case, audio features are more discriminative to detect the nature of a person’s voice.

The kappa coefficients are calculated for the confusion matrices in Table 6.1. For TVHID and Parliament datasets, the kappa coefficients are 0.7119 and 0.9000, respectively. This shows that there is a strong agreement that the observed CRs are better than those of a chance classifier.

6.2.4 Comparison with State-of-the-art Methods for Human Interaction Recognition

In this experiment, the proposed audio-visual recognition system i.e., LRGS-STIP+SDTD, is compared with some state-of-the-art methods [1], [2], [129], [130], [160], [161], which use different type of features (i.e., visual and audio-visual). The CRs of the proposed and other methods are given in Table 6.2. The CRs of the other methods are directly taken from the references shown in the table.

For TVHID dataset, the LRGS-STIP+SDTD approach achieves a CR of 78.5%, and it outperforms most of the other methods. The p-values from the comparison of the proposed LRGS-STIP+SDTD with other methods are given in the last row of Table 6.3. The CRs of the other methods are directly taken from the references shown in the table. The LRGS-STIP+SDTD is compared (using the Friedman’s test) to only those methods which provide CRs of individual classes.
Table 6.2: Average CR ± std (in percent) of the proposed LRGS-STIP+SDTD and other methods for TVHID and Parliament datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>Features type</th>
<th>TVHID</th>
<th>Parliament</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAVAR [2]</td>
<td>Audio-visual</td>
<td>81.3 ± 2.3</td>
<td>97.6 ± 1.0</td>
</tr>
<tr>
<td>Li et al. [160]</td>
<td>Visual</td>
<td>68.0 ± 2.7</td>
<td>-</td>
</tr>
<tr>
<td>Yu et al. [161]</td>
<td>Visual</td>
<td>66.2 ± 2.7</td>
<td>-</td>
</tr>
<tr>
<td>Patron et al. [129]</td>
<td>Visual</td>
<td>54.7 ± 2.9</td>
<td>-</td>
</tr>
<tr>
<td>Marin et al. [1]</td>
<td>Audio-visual</td>
<td>54.5 ± 2.9</td>
<td>-</td>
</tr>
<tr>
<td>Vrigkas et al. [130]</td>
<td>Visual</td>
<td>-</td>
<td>85.5 ± 2.3</td>
</tr>
<tr>
<td>LRGS-STIP+SDTD</td>
<td>Visual</td>
<td>73.0 ± 2.6</td>
<td>88.6 ± 2.1</td>
</tr>
<tr>
<td>LRGS-STIP+SDTD</td>
<td>Audio-visual</td>
<td>78.5 ± 2.4</td>
<td>93.4 ± 1.6</td>
</tr>
</tbody>
</table>

for the dataset. A p-value of 0.03 (less than 0.05) is obtained when comparing all the methods in Table 6.3 simultaneously. This shows that all the methods (including the proposed one) have significantly different performances. The p-values obtained after individual comparison of the proposed method with the methods in [129] and [161] are less than 0.05, which means there is significant difference between the CRs. Although the CR of the method in [2] is higher than our method, the difference between the CRs is not statistically significant (p-value = 0.32, greater than 0.05).

Table 6.3: The Friedman’s test p-value for the proposed LRGS-STIP+SDTD in comparison with other methods for TVHID dataset.

<table>
<thead>
<tr>
<th>Scenes</th>
<th>LRGS-STIP+SDTD</th>
<th>SAVAR [2]</th>
<th>Patron et al. [129]</th>
<th>Yu et al. [161]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand shake</td>
<td>76</td>
<td>87</td>
<td>41</td>
<td>66</td>
</tr>
<tr>
<td>High five</td>
<td>84</td>
<td>56</td>
<td>43</td>
<td>58</td>
</tr>
<tr>
<td>Hug</td>
<td>78</td>
<td>88</td>
<td>66</td>
<td>71</td>
</tr>
<tr>
<td>Kiss</td>
<td>76</td>
<td>94</td>
<td>69</td>
<td>70</td>
</tr>
<tr>
<td>p-value</td>
<td>LRGS-STIP+SDTD vs.</td>
<td>0.317</td>
<td>0.046</td>
<td>0.046</td>
</tr>
</tbody>
</table>

For Parliament dataset, the proposed LRGS-STIP+SDTD approach achieves a CR of 93.4%, which is second highest in comparison with other methods. The p-values from the comparison of the proposed LRGS-STIP+SDTD with other methods are given in the last row of Table 6.4. The CRs of the other methods are directly taken from the references shown in the table. A p-value of 0.44 (greater than 0.05) is obtained when comparing all the methods in Table 6.4 simultaneously.
Also, the $p$-values obtained after individual comparison of the proposed method with the methods in [2] and [130] are greater than 0.05. This means that all the methods have similar performances. Only SAVAR [2] method, which extracts more features such as head orientation and proxemic, outperforms our proposed method. Although the CR of SAVAR method is higher than our method, the difference between the CRs is not statistically significant ($p$-value = 0.16, greater than 0.05).

Table 6.4: The Friedman’s test $p$-value for the proposed LRGS-STIP+SDTD in comparison with other methods for Parliament dataset.

<table>
<thead>
<tr>
<th>Scenes</th>
<th>LRGS-STIP+SDTD</th>
<th>SAVAR [2]</th>
<th>Vrigkas et al. [130]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Friendly</td>
<td>93</td>
<td>93</td>
<td>100</td>
</tr>
<tr>
<td>Aggressive</td>
<td>96</td>
<td>100</td>
<td>61</td>
</tr>
<tr>
<td>Neutral</td>
<td>91</td>
<td>100</td>
<td>96</td>
</tr>
<tr>
<td>$p$-value</td>
<td>LRGS-STIP+SDTD vs.</td>
<td>0.157</td>
<td>0.564</td>
</tr>
</tbody>
</table>

6.3 Violent Scene Detection

6.3.1 Related Work

The task of violent scene detection (VSD) has been studied before, especially in the video surveillance domain. In case of movies, the VSD task is significantly different where so many audio and visual effects are involved due to high editing. We focus on VSD in movies and user generated videos uploaded on internet (e.g., YouTube).

The VSD task becomes complex due to the subjective and ambiguous definition of violence. This causes researchers difficulty in terms of working on a common ground [162]. Some of the interpretations of violence refer to scenes that contain blood, gunshots, and explosions [163], [164], actions that involve person-to-person threats and physical harm [10], and scenes of fights among people (regardless of number of individuals involved and context) [165], [166]. These different interpretations lead to different techniques for the VSD task, which makes it difficult to conduct a comparative study. Furthermore, the presence of multiple modalities and unknown duration of events complicate the problem further.
6.3. Violent Scene Detection

The different approaches can be categorized in terms of feature types extracted for classification (i.e., audio, visual, and textual). For example, in [167] and [168], the authors use single modality (i.e., audio events) and extract different audio features including zero-crossing and energy entropy. Many researchers, on the other hand, have been interested in combining both auditory and visual modalities. The combined use of audio (e.g., chroma, spectrogram, and MFCC) and visual features (e.g., motion based variance, motion of people, and average motion) produced some good results [10]. In [11], the authors perform a modified probabilistic Latent Semantic Analysis (pLSA) based violence detection from audio cues and visual information by exploiting different concepts (including explosion, motion, blood, and flame). Many other methods have been proposed that merge the two modalities of audio and visual information for the VSD task [5]–[9]. Other than audio-visual features, some authors also exploited the use of textual information [9], [169].

MediaEval has been providing a benchmark for the VSD task in movies since year 2011 [170]. The Affect Task of MediaEval provides researchers a common ground to work on this problem and compare their algorithms [171]. In MediaEval 2014, many teams participated for the VSD task. In [172], the authors use deep neural networks along with support vector machines and extract different audio-visual features (i.e., MFCC, dense trajectories, and STIPs). This method performed best of all on one of the two VSD sub-tasks (i.e., violence detection in Hollywood movies). In [173], a set of mid-level concepts is predicted from many low level audio and visual features and then the features and concept predictions are fused to detect the violent scenes. This approach outperformed all the other methods on the second VSD sub-task (i.e., violence detection in user generated videos from YouTube). The most common features used by most of the participating teams are MFCC (audio) and dense trajectories (visual+temporal) [171].

6.3.2 Experimental Method

We test our audio-visual recognition system (RDT+SDTD) for violent scene detection using the MediaEvalv2014 VSD dataset (VSD2014) [171]. The VSD2014 dataset contains three subsets: Development, Test, and Generalization subsets. The
6.3. Violent Scene Detection

![Sample video frames from the MediaEval VSD2014 dataset](image)

*Figure 6.1: Sample video frames from the MediaEval VSD2014 dataset [171].*

*Development* and *Test* subsets consist of Hollywood movies, and the *Generalization* subset contains video clips from YouTube. There are twenty-four movies in the *Development*, seven movies in the *Test*, and eighty-six clips in the *Generalization* subsets, with average violence rate of 12.35%, 17.18%, and 31.69%, respectively. Frame level binary annotations are provided for all the scenes. The violent scenes are identified by their start and end frames. Fig. 6.1 shows some violent scenes (e.g., explosion, fights, gun-shot, screaming, and war violence) from the VSD2014 dataset.

To be consistent with the participating teams for the VSD Affect task at MediaEval 2014, we perform the same violence detection task and use the same evaluation protocol. The VSD Affect task aim to auto-detect the violent video segments in movies by indicating their start and end frames. With this information, it is easy to make a summarized video containing violent scenes for parental guidance. For evaluation, a modified version of the mean average precision (MAP): dubbed
6.3. Violent Scene Detection

MAP2014, is used [171]. The MAP2014 measure considers as a hit only predicted segments that overlap by more than 50% with their corresponding ground truth segments. If there are multiple hits on the same ground truth, only one true positive is counted and the rest are ignored.

In our audio-visual recognition system, the audio features are extracted using MFCC and visual features are extracted using the RDT method. The audio-visual features are then represented by the SDTD model. The implementation details of audio-visual features extraction and representation are given previously in Sections 3.3.4.2, 3.2.5.2, and 4.4.2. The videos in the test subsets (i.e., Test and Generalization) are subdivided into 75 frames clips. For the desired segment level prediction output, the continuous clips are merged to get a single video segment if they are all classified as violent or non-violent.

6.3.3 Classification results for Violent Scene Detection

The classification results for the proposed audio-visual recognition system are presented in this section. The classification results are obtained using the best configuration of our SDTD model, as concluded in Chapter 4. For this purpose, SDV is used for feature encoding (with dictionary size of 500), HOOI algorithm is used for tensor decomposition, Fisher ranking is used for feature selection, and ELM is used for classification. The MAP2014 scores as a function of number of features are given in Fig. 6.2 for the two VSD2014 subsets: Test (Hollywood) and Generalization (YouTube). The highest MAP2014 score of 61.6% is achieved for the Test (Hollywood) subset using 8,000 features. The highest MAP2014 score of 68.4% is achieved for the Generalization (YouTube) subset using 7,500 features.

6.3.4 Comparison with State-of-the-art Methods for Violent Scene Detection

The proposed audio-visual recognition system i.e., RDT+SDTD, is compared with several methods presented for the VSD Affect task at MediaEval 2014. The participating teams include RECOD [169], FUDAN [172], FAR [173], NII-UIT [174], MIC-TJU [175], VIVOLAB [176], TUB-IRML [177], and MTMDCC [178]. The MAP2014 scores of the RDT+SDTD and other methods for the Test (Hollywood)
and Generalization (YouTube) subsets are given in Table 6.5. The MAP2014 scores of the other methods are directly taken from the references shown in the table. For the Test (Hollywood) subset, the RDT+SDTD achieves a MAP2014 score of 61.6%, and it outperforms all other methods except for FUDAN which has a score of 63.0%. For the Generalization (YouTube) subset, the RDT+SDTD achieves a MAP2014 score of 68.4%, and it outperforms all other methods including FUDAN.

The Friedman’s test is not feasible for the statistical significance comparison of different methods tested on VSD2014 dataset. The reason is that VSD2014 is not a multi-class dataset like Maryland, YUPPEN, or UCF, where the CRs (data) of individual categories are used to compute the Friedman’s $p$-values for comparison between two methods. If the MAP2014 scores in Table 6.5 are used to calculate the Friedman’s $p$-values, the results may look unrealistic because there are not enough subsets or data. For example, a $p$-value of 0.16 is obtained when comparing the RDT+SDTD with VIVOLAB, which indicates that the difference between the MAP2014 scores of the two methods is not significant. Since this does not give us a true comparison, the Friedman’s test is not performed for VSD2014 dataset to compare different methods.
### 6.4 Chapter Summary

In this chapter, we analyze the performance of our proposed audio-video recognition system for the tasks of human interaction recognition and violent scene detection. The audio features are extracted using Mel-frequency cepstral coefficients. To extract visual features, the proposed LRGS-STIP and RDT methods are used for human interaction recognition and violent scene detection datasets, respectively. The extracted audio-visual features are then represented through the SDTD model for classification. Audio and visual features together provided better classification accuracy than visual only features. The results show that the proposed audio-visual recognition system outperforms most of the other methods for tasks of human interaction recognition and violent scene detection.

<table>
<thead>
<tr>
<th>Team/Method</th>
<th>Test (Hollywood)</th>
<th>Generalization (YouTube)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUDAN [172]</td>
<td>63.0 ± 2.8</td>
<td>60.4 ± 3.5</td>
</tr>
<tr>
<td>NII-UIT [174]</td>
<td>55.9 ± 2.9</td>
<td>-</td>
</tr>
<tr>
<td>FAR [173]</td>
<td>45.1 ± 2.9</td>
<td>66.4 ± 3.3</td>
</tr>
<tr>
<td>MIC-TJU [175]</td>
<td>44.6 ± 2.9</td>
<td>56.6 ± 3.5</td>
</tr>
<tr>
<td>RECOD [169]</td>
<td>37.6 ± 2.8</td>
<td>61.8 ± 3.4</td>
</tr>
<tr>
<td>VIVOLAB [176]</td>
<td>17.8 ± 2.2</td>
<td>43.0 ± 3.5</td>
</tr>
<tr>
<td>TUB-IRML [177]</td>
<td>17.2 ± 2.2</td>
<td>51.7 ± 3.5</td>
</tr>
<tr>
<td>MTMDCC [178]</td>
<td>2.6 ± 0.9</td>
<td>-</td>
</tr>
<tr>
<td>RDT+SDTD</td>
<td>61.6 ± 2.8</td>
<td>68.4 ± 3.3</td>
</tr>
</tbody>
</table>
Chapter 7

Conclusion

Chapter contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1</td>
<td>Research summary</td>
<td>104</td>
</tr>
<tr>
<td>7.2</td>
<td>Future work</td>
<td>105</td>
</tr>
<tr>
<td>7.3</td>
<td>Conclusion</td>
<td>105</td>
</tr>
</tbody>
</table>

Audio-visual recognition systems have been proposed for automatic recognition of dynamic scenes, human actions, events, and human interactions. Many applications include automatic surveillance, human-computer interaction, video indexing and retrieval, games, virtual reality, and robot navigation. Existing recognition systems show some limitations of visual feature extraction and global feature representation in videos. We propose to extract salient and discriminative visual features in presence of camera motion, and present a tensor based global feature representation model to retain the spatio-temporal structure among features. Improving individual components of a recognition system (i.e., visual feature extraction and global feature representation) leads to better classification accuracy for video recognition.

This chapter is organized as follows: Section 7.1 summarizes the research contributions of the thesis; Section 7.2 outlines the future research directions; Section 7.3 draws conclusion for the thesis.
7.1 Research summary

The research activities have been documented in several chapters of the thesis. They are listed and summarized as follows:

1. We provided a literature review of audio-visual recognition systems and their individual components: audio-visual feature extraction, global feature representation, and video classification.

2. We proposed a new method for visual feature extraction called refined dense trajectories. Salient interest points are detected in a region of interest where the motion is the most discriminative. The refined dense trajectories method provides salient trajectories by discarding unnecessary and noisy interest points.

3. We presented a novel spatio-temporal interest point detector based on a low-rank and group-sparse matrix approximation. The detector yields a set of salient spatio-temporal interest points that is neither too dense nor too sparse. The detector incorporates long-term temporal interactions to detect spatio-temporal interest points, which represent the best key points in motion areas.

4. We integrated a short-window video stabilization in the above visual feature extraction methods to handle camera motion. The global motion is compensated by realigning of the video frames during interest point detection and trajectory formation. This yields a stabilized set of interest points and trajectories, which is not affected due to the camera motion and dynamic background.

5. We proposed a unique super descriptor tensor decomposition model for global representation of audio-visual features from multiple descriptors and modalities. Discriminative features are obtained for classification through decomposition of a tensor-based model followed by a feature ranking. This retains the spatio-temporal structure among features from multiple descriptors and modalities.
7.2 Future work

Possible research directions can be summarized as follows:

1. The super descriptor tensor decomposition model is capable of accommodating datasets that contain multiple modalities (e.g., text, audio, visual, depth, and 3D shape). This capability of the super descriptor tensor decomposition model can lead to an extension of the model towards large-scale multi-modal datasets.

2. Tensor decomposition is inherently a computationally intensive process. A possible research direction is to explore graphic processing units for fast computation to accommodate real-time video processing.

3. A massive amount of video data is generated every day, e.g., thousands of videos are uploaded on social media and video streaming websites daily. Reducing the video annotation cost is in high demand. For this purpose, active learning can be incorporated with the super descriptor tensor decomposition model to automatically annotate the videos.

7.3 Conclusion

In this research project, firstly, a new method called refined dense trajectories was proposed for visual feature extraction. This method was compared with the widely used dense trajectories method. Our method outperforms the dense trajectories method in terms of visual analysis, classification accuracy, and computation time. Secondly, a novel spatio-temporal interest point detector was presented based on a low-rank and group-sparse matrix approximation. To handle camera motion, a short-window video stabilization was presented, which compensates for global motion by realigning of the video frames, during interest point detection and trajectory formation. The proposed detector was compared with some existing spatio-temporal interest point detectors. Our detector outperforms the other detectors in terms of valid interest point detection and classification accuracy, with and without adding global motion compensation. Thirdly, a unique
super descriptor tensor decomposition model was presented for feature representation from multiple descriptors and modalities. In our model, the super descriptor vector coding yields best classification results for a small dictionary size. In addition, the higher-order orthogonal interactions along with Fisher ranking provides the best results for a small number of features used for classification. The super descriptor tensor decomposition model was compared with existing global feature representation methods including bag-of-words and super vector based models. Our model outperforms all other methods in terms of classification accuracy. We evaluated different classifiers for our proposed audio-visual recognition system. The extreme learning machines classifier provides the best classification results. Lastly, the proposed visual and audio-video recognition systems were tested on multiple visual and audio-visual datasets, for the tasks of dynamic scene recognition, action recognition, human interaction recognition, and violent scene recognition. The reliability of the obtained classification results for the proposed recognition systems was measured using the Cohen’s kappa coefficient. The classification results of the proposed recognition systems were compared with other methods using the Friedman’s test. From the comparison, the proposed recognition systems either outperform or give comparable classification results to the state-of-the-art methods for visual and audio-visual recognition. In future, the proposed systems are to be employed for large-scale multi-modal datasets. In addition, graphic processing units are to be explored for fast processing. Furthermore, active learning is to be incorporated for automatic annotation to accommodate large-scale datasets.
Appendix

A.1 Derivation of Equation (4.2)

The gradient of the log-likelihood of $p(x_i)$ w.r.t mean $\mu_k$ for a component $k$,
\[
\frac{\partial}{\partial \mu_k} \ln p(x_i) = \frac{1}{p(x_i)} \frac{\partial}{\partial \mu_k} \{w_k p_k(x_i)\}. \tag{A.1}
\]
The R.H.S of (A.1) can be expressed as
\[
\frac{\partial}{\partial \mu_k} \ln p(x_i) = \frac{w_k p_k(x_i)}{p(x_i)} \left[ \frac{1}{w_k p_k(x_i)} \frac{\partial}{\partial \mu_k} \{w_k p_k(x_i)\} \right]. \tag{A.2}
\]
Using Bayes’ rule to obtain the posterior $p_k^+(x_i)$ in (A.2) gives
\[
\frac{\partial}{\partial \mu_k} \ln p(x_i) = p_k^+(x_i) \left[ \frac{\partial}{\partial \mu_k} \ln \{w_k p_k(x_i)\} \right]. \tag{A.3}
\]
Taking the gradient of the log-likelihood of $w_k p_k(x_i)$ leads to:
\[
\frac{\partial}{\partial \mu_k} \ln \{w_k p_k(x_i)\} = \frac{\partial}{\partial \mu_k} \{\ln w_k + \ln N(x_i; \mu_k, \sigma_k)\} \tag{A.4}
\]
\[
= \frac{\partial}{\partial \mu_k} \ln N(x_i; \mu_k, \sigma_k).
\]
Applying the log to the multi-variate Gaussian $N(x_i; \mu_k, \sigma_k)$ in (A.4) gives
\[
\frac{\partial}{\partial \mu_k} \ln \{w_k p_k(x_i)\} = \frac{\partial}{\partial \mu_k} \left\{ \frac{1}{2} (x_i - \mu_k)^T \sigma_k^{-1} (x_i - \mu_k) \right\}. \tag{A.5}
\]
From the Equation (86) of the matrix cookbook by Peterson et al. [179], we can simplify (A.5) as
\[
\frac{\partial}{\partial \mu_k} \ln \{w_k p_k(x_i)\} = \frac{-1}{2} \{ -2 \sigma_k^{-1} (x_i - \mu_k) \} \tag{A.6}
\]
\[
= \sigma_k^{-1} (x_i - \mu_k).
\]
Finally, substituting (A.6) into (A.3) leads to:

\[
\frac{\partial}{\partial \mu_k} \ln p(x_i) = p_k^+(x_i) \sigma_k^{-1}(x_i - \mu_k).
\]  

(A.7)
References


References


