M.S. Dissertation

Machine Vision for Human Activity Recognition: Features & Algorithms

행동인식을 위한 머신비젼기술: 특징 및 알고리듬 연구

By

Tushar Sandhan

August 2014

SCHOOL OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Machine Vision for Human Activity Recognition: Features & Algorithms

행동인식을 위한 머신비젼기술:

특징 및 알고리듬 연구

지도 교수 최 진 영

이 논문을 공학박사 학위논문으로 제출함

2014 년 8 월

서울대학교 대학원

전기컴퓨터공학부

투샤르 산단

투샤르 산단의 공학박사 학위논문을 인준함

2014 년 8 월

위 원 장 조 남 익 (인)

부위원장 최 진 영 (인)

위 원 오 성 회 (인)

Abstract

Human Activity Recognition (HAR) is a multifaceted aspect of computer vision and machine learning, which encompasses group activity pattern discovery, interpersonal interaction analysis, human gesture and action recognition. It has proliferating demands from wide applications, such as visual surveillance and security, entertainment, healthcare systems, video indexing, human-computer interaction and video retrieval. So over the last decade, a diversity of approaches has been developed to investigate the HAR. We overcome their limitations by proposing new robust features and the algorithms to build the unified HAR framework.

Features play a vital role in HAR. Global features are generated using the entire video sequence while ignoring explicit temporal information but they capture the oriented and holistic underlying patterns. We found that HAR can be improved by fusing extra temporal information with global representation. So following this vein, we propose the new mid-level features (*frequencygrams* and *spatiograms*) by analyzing dynamics of the motion histograms in frequency and spatio-temporal domain; and the new high-level features (*abstracted radon profiles*) by considering whole oriented information of the action silhouettes. They are robust to camera motions and small occlusions and provide a discriminative representation for reciprocating motions. They are used for supervised HAR through the proposed *graph pyramid*, a hierarchical graph analysis algorithm. In *graph pyramid* classification algorithm, we construct the graph of an entire action class by representing each motion sequence as a node. Training embeds the class specific information in the graph edges via our modified quadratic-Chi distance. The algorithm makes it possible to uncover the hidden subtleties of the action family by considering interactions among the neighborhood nodes to the query.

Optical flow is the basis to describe motion sequence, however it is in raw form may be of no use due to its susceptibility to background noise, camera motions and scale changes. But the constructed features from raw data, encapsulate the underlying dynamics of the activity, so they play an important role here. Using raw optical flow we also propose the low-level statistical motion features (viz., *circulation*, *motion homogeneity*, *motion orientation* and *stationarity*) to readily capture the pixel level motion information. Then we use these features for unsupervised abnormal activity recognition by the proposed *proximity clustering* algorithm. The key idea behind it is that the normal events occur more frequently than the abnormal ones. It clusters the normal events in the proposed feature space and outliers are designated as abnormal events. It works on proximity principal and does not require to specify the number of (normal events) clusters.

In HAR domain, some action classes have very less training examples. Without dataset rebalancing, the learning algorithm will encounter extremely low minority class samples therefore it gets biased towards the majority class. Hence properly handling the imbalanced dataset is a crucial issue. To address it, we propose the G-SMOTE algorithm by employing bootstrapping with simultaneous oversampling of minority class and undersampling of majority class to build the ensemble of classifiers. G-SMOTE is an improvement to the existing synthetic minority oversampling technique. Its extensive evaluation on several highly imbalanced datasets has produced the highest recognition results.

In case of traffic scenario, we are the first to implement the time series embedding framework to solve the data scarcity problem for traffic activity recognition. Using multi-task learning framework, we learn all activity classifiers simultaneously by exploiting correlations among different motion patterns. We have improved the traffic pattern recognition performance on all four public domain datasets by several magnitude as compared to the state-of-the-art approaches.

Machine vision becomes blind in case of dark illumination conditions, occlusions or in the areas outside the camera view. The use of audio information along with the video can enhance the performance of the HAR system for better understanding of the underlying scene. So we propose the *audio bank*, a new high-level representation of an audio, for audio activities recognition. It is comprised of distinctive audio detectors representing each audio class in the frequency-temporal space. It produces superior features as compared to low-level features in discriminating audio events by accumulating responses of all bank detectors into one vector. Feature stability over the bank size and high recognition performance using several classifiers show the effectiveness of the proposed method.

Keywords: activity recognition (action; gesture; abnormal events), features, hierarchical graph analysis, proximity clustering, imbalanced dataset handling

Student ID: 2012-23964

Name: Tushar SANDHAN

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivations and Challenges

Machine can see through the camera eye!

We are drowning in the deluge of video data that are being captured worldwide by the cameras. At the same time we are starving for the meaningful and concise information from that data. Surveillance cameras are mounted for monitoring surrounding events and human activities, whose video data is being assessed by another human observer. However, as a number of cameras and an observation duration increases, events monitoring by human operators becomes increasingly difficult and error-prone. In order to assist operators, many automatic surveillance systems have been proposed [1], [2]. These systems increase the robustness of video monitoring. The ever increasing need for intelligent surveillance in public security domain, makes automated video surveillance systems to be unavoidable.

Human Activity Recognition (HAR) is not only related to surveillance applications, but it is also a multifaceted aspect of the computer vision and the machine learning. HAR encompasses group activity recognition, human gesture

(a) (b) (c) (d) (e)

Figure 1.1: Challenges in HAR. Video frames from various activity datasets are used here. (a) Actor view point and scale changes in the same *golfing* action. (b) 1^{st} row shows the illumination change in *handwaving* action and 2^{nd} row shows the change in direction of *boxing* action execution. (c) 1^{st} row is the frame from *running* action video with moving background (camera shakes) and 2^{nd} row shows its optical flow, which is erroneous and occupied over entire frame. (d) Different scenarios for abnormal activities show completely different motion patterns. Only small of those scenarios are available for training. Hence scarcity of training data. (e) Here *running*, *jogging* and *walking* are completely different activities but their execution speed depends upon actor and might be quite similar.

recognition, interpersonal interaction analysis and human action recognition. It has proliferating demands from wide applications, such as entertainment, healthcare systems, video indexing, human-computer interaction, visual surveillance-security, enhancing computer graphics and video retrieval. So over the last decade, a diversity of approaches has been developed to investigate the HAR. Most of these approaches are tackling the aspects of HAR (viz., individual action, gesture, traffic patterns, group activities related to surveillance, abnormal events, etc.) independently. There is lack of robust and unified framework to handle the challenges in HAR as shown in Fig.1.1.

The first challenge is to achieve the real time processing (25 to 30 frames per second) of the enormous raw data for pattern recognition because usual video frame resolution ranges from 320×180 to 1280×720 pixels and each color pixel comprises of 24 bits data representation. Other issues that need to be addressed in HAR are summarized as follows,

- Same activity looks quite distinct when observed from different view point. In Fig.1.1(a), same *golfing* looks different for machine from different views.

- Scale changes: a person far way from camera looks smaller than the one closer to camera (Fig.1.1(a)). If the features are not normalized appropriately then small scale actions get dominated by large scale actions.
- Moving background or camera shakes make motion to appear over entire frame (see Fig.1.1(c)), and it makes difficult for machine to find important motion patterns (only foreground motions) necessary for HAR.
- Sudden illumination changes produces extra noisy motion vectors. In the Fig.1.1(b), all 4 video frames are from same KTH dataset but they show quite different illumination conditions.
- Continuous dark illumination conditions, occlusions make the machine blind.
- Different person may perform different activities with the same speed. In Fig.1.1(e) *running*, *jogging* and *walking* are completely different activities but *running* speed of one actor might be quite similar with *jogging* speed of another actor.
- Action direction change. In Fig.1.1(b) bottom row shows the *boxing* in different directions. More importantly HAR is challenging because each person has his own style of acting!

1.2 Contents of Research

The thesis is mainly focused on developing the overall robust framework for human activity recognition by proposing the new features and new algorithms to cope with the challenges in HAR mentioned in section 1.1. Its brief overview in terms of contributions and contents of research is shown in Fig.1.2.

Video data in its raw form has very high dimensionality and it is difficult to process in real time for recognizing human activity patterns. So we propose the

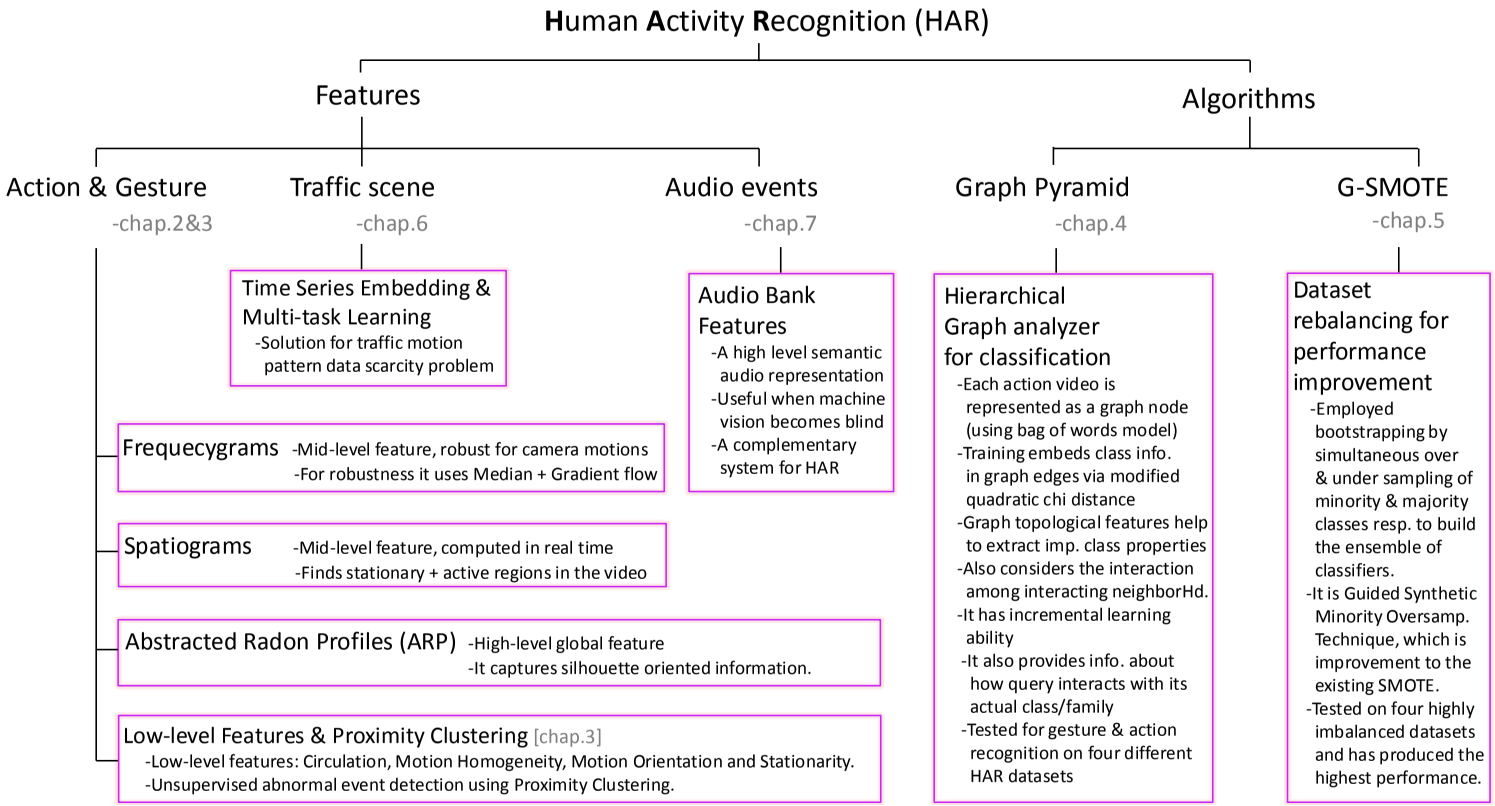


Figure 1.2: Brief overview of the contributions in the thesis (A complete framework for HAR).

features, which can encapsulate the underlying dynamics of the activity. We propose the *low-level features* (viz., Circulation, Motion Homogeneity, Motion Orientation and Stationarity), which readily capture the pixel level motion information. For unsupervised abnormal activity recognition using these low-level features, we propose the *Proximity clustering algorithm*. For robust activity recognition we propose the *mid-level features* (viz., Frequencygrams and Spatiograms) and the *high-level features* (viz., Abstracted Radon Profiles). We are the first to implement the time series embedding framework to solve the data scarcity problem for traffic activity recognition. We also propose the *Graph Pyramid*, a hierarchical graph analysis algorithm for general inference and pattern recognition purpose and we validate it extensively for HAR. We also make use of acoustic information for offering more robustness to HAR, by proposing the *Audio Bank*, a high-level semantic feature representation.

The proposed unified framework for HAR has been extensively tested on various datasets, viz., HWU human gesture [3], Weizmann action [4], UMN abnormal activity [5], KTH action [6], UCF-Sports [7], UPC-TALP audio [8], QMUL traffic [9], MIT [10], NGSIM [11] and Wide Interaction (WI) our own traffic dataset.

In the following sections we present brief overview of the contributions (proposed features and algorithms) to build the robust and unified HAR framework.

1.2.1 Mid-Level and High-Level Features

Chapter 2 describes the proposed mid-level (*frequencygrams*, *spatiograms*) and high-level (*abstracted radon profiles*) features. They are robust to camera motions and provide a compact and discriminative representation for reciprocating motions by preserving comprehensive temporal information of an activity. The extensive experimental results on the four benchmark human activity datasets demonstrate their effectiveness as well as generality.

1.2.2 Low-Level Features and Proximity Clustering Algorithm

Uncovering the hidden subtleties and irregularities of the events in the video sequence, is the key issue for automatic video surveillance and chapter 3 deals with that. The proposed motion features, viz., *circulation*, *motion homogeneity*, *motion orientation* and *stationarity* try to extract vital motion information from the video. Then we propose the *proximity clustering* algorithm, which works in the above feature space to find the abnormal events by unsupervised grouping of the frequently occurring features. The key idea here is, the occurrence of abnormal events is rare while the frequently occurring events become normal in general human perception. This method has an advantage of incremental learning, i.e. it learns the new normal events in an unsupervised manner.

1.2.3 Graph Pyramid: A hierarchical graph analysis framework

In chapter 4, we propose a graphical model for pattern recognition from data. We extensively evaluate it for HAR. Unlike other methods, we construct the graph of the entire action class by representing each video as a node. Training embeds the class specific information in the graph edges via our modified quadratic-Chi distance. Hierarchical graphical analysis makes the inference robust as it also considers the interactions among the neighborhood nodes to the query.

1.2.4 Handling Imbalanced Datasets: G-SMOTE Algorithm

Occurrence of high class imbalance in the HAR domain a direct result of rarity of interesting events, which results in skewed datasets. Without dataset rebalancing, the learning algorithm will encounter extremely low minority class samples therefore it gets biased towards the majority class in the classification tasks. Hence properly handling the imbalanced dataset is a crucial issue in the pattern

recognition domain. So we have proposed the generalized G-SMOTE dataset rebalancing algorithm in chapter 6. The proposed framework is evaluated on four highly imbalanced datasets, which has produced the state-of-the-art performance.

1.2.5 Traffic Activities: Features and Mulit-task Learning

Chapter 7 proposes an efficient feature sampling and multi-task learning scheme for traffic scene analysis, where all classifiers are trained simultaneously by exploiting the correlations among different motion patterns. We make feature descriptors by high dimensional embedding of the time series data for traffic patterns. They preserve detailed spatio-temporal information of the underlying event. Experimental results on surveillance datasets, show great improvement in the performance, importance of joint feature selection and fast incremental learning ability of the proposed method.

1.2.6 Audio Activities Recognition

Machine vision becomes blind in case of dark illumination conditions, occlusions or in the areas outside the camera view. The use of audio information along with the video can enhance the performance of the HAR system for better understanding of the underlying scene. So we propose the *audio bank*, a new high-level representation of an audio, for audio activities recognition in chapter 5. It produces superior features as compared to low-level features in discriminating audio events. Feature stability over the audio bank size and high recognition rate using several classifiers show its effectiveness. Finally chapter 8 gives broad concluding remarks about the unified HAR framework.

Chapter 2

Mid-Level and High-Level Features

2.1 Introduction and Prior Work

As alluded to the importance of Human Activity Recognition (HAR) in chapter 1, it is multifaceted aspect of the computer vision and machine learning. So over the last decade, a diversity of approaches has been developed to investigate the HAR. Since appearance and motion dynamics serve as the important cues for it, the prior work can be broadly categorized as appearance-based [12, 13, 14] and motion-based [15, 16, 17, 18] approaches. The former characterizes the video by leveraging the contextual information via visual features [19, 20]. Their major problem is that they discard the inherent temporal information of the human activity. Whereas later models the activities with state-space or dynamic models and casting HAR as a temporal classification task. Though they keep account of the sufficient temporal information using detailed statistical modeling, they have high computational complexity, which is a bottleneck for mobile platforms and

real time surveillance applications where computational resources are limited. We circumvent these issues by analyzing the distribution of motion patterns in the frequency and the spatio-temporal domain.

In motion-based HAR approaches, as features have the capability to encapsulate underlying dynamics of the activity, they plays a vital role here. Local features like silhouette, gesture, human body part joints, dense trajectories [21], local space-time features [19, 22] and dense 3D gradient histograms [23] are computationally expensive and limited in the amount of activity semantics they can capture. In addition they require salient detection-tracking and their performance degrades during camera movements, occlusions, viewpoint and illumination changes. They often produce a representation with inadequate discriminative power for the large datasets. On the contrary, global features like motion energy images [24], Radon profiles [25] over silhouettes and motion history images (MHI) [26] are generated using the whole motion sequence so they ignore the explicit temporal information but captures the oriented and holistic underlying patterns. Though they bear some action semantics due to their global nature, they lack of sufficient temporal information as the entire motion history is accumulated in the final global feature. For instance, it is difficult to distinguish '*running*' from '*walking*' using just MHIs and they are incapable of handling reciprocating actions (*walking* leftward vs rightward) [27]. So HAR can be improved by fusing extra temporal information with global features. Thus our work follows this vein.

To address these issues, we propose the simple but effective mid-level features called 'Frequencygrams' and 'Spatiograms', which incorporate the comprehensive temporal information by analyzing the activity in the frequency and the spatio-temporal domain respectively (section 6.2). We also propose the high-level features called 'Abstracted Radon Profiles (ARP)', which captures the global spatial information from the human silhouette. ARP features have been extensively

tested on large fingerprint dataset FVC-2006 in our work [28]. So in this chapter we exclude the experimental results for high-level ARP features but give the detailed HAR results for the proposed mid-level features. These oriented holistic features provide a natural and discriminative representation for reciprocating motions using directional and spatio-temporal motion information. The proposed approach has an advantage of being robust to camera motions, occlusions and it does not necessitate extra video preprocessing, person detection and tracking overload unlike other low-level features. To improve the HAR rate using all related features, we first approximate their conditional probability distributions for each activity class independently and then make inference about test sample by maximizing the collective likelihood (section 2.5). The proposed framework is validated on four benchmark human activity (human gesture, action and abnormal activities) datasets (sec. 6.5) and finally sec. 6.6 concludes this chapter.

2.2 Synopsis

If features can encapsulate the underlying dynamics of the activity then they play a vital role in human activity recognition. But most of the existing features fail to capture the detailed temporal information of an activity in the video. So we propose new mid-level (Frequencygrams, Spatiograms) and high-level (Abstracted Radon Profiles) features by analyzing motion dynamics and appearance in the spatio-temporal domain. They are computationally efficient and avoid overload of extra video preprocessing. Being robust to camera motions, they also provide a natural, compact and discriminative representation for reciprocating motions by preserving comprehensive temporal information of the activity sequences. The extensive experimental results on the four benchmark human activity datasets demonstrate the effectiveness as well as generality of the proposed framework.

2.3 Mid-Level Feature Construction

Important clues for recognizing activity in the video are obtained by answering, 'how' and 'where' the activity is being performed over the course of time. The answer to first question discovers the underlying motion dynamics of the activity, whereas the answer to second one reveals the spatial distribution of the activity occurrences. The proposed features encapsulate these answers and extract vital information from the video necessary for activity recognition.

2.3.1 Histogram Flow Pattern

Optical flow is the basis to describe motion sequence, however it is in raw form may be of no use due to its susceptibility to background noise, camera motions, changes in scale and motion direction. The success of histograms of features [29], is well known in object recognition community. Hence analyzing distribution of the optical flow profile would be helpful to improve activity recognition by avoiding the above issues.

Each video $\mathcal{V}$, is analyzed via set of N_B blocks, where each block is having N_F number of frames. Thus it is represented as $\mathcal{V} = \{b_1, b_2, \cdots, b_{N_B}\}$, $b_t \in \mathcal{V}$ and $|b_t| = N_F$. At time t, the optical flow vector $\bar{\nu}_t = [u, v]^T$, within each b_t is binned according to its primary angle with horizontal axis and weighted by its magnitude (see Fig.2.1). That is, $\bar{\nu}_t$ from b_t, in the range $\frac{\pi(i-1)}{N_b} \leq \frac{\pi}{2} + tan^{-1}(\frac{u}{v}) < \frac{\pi i}{N_b}$, will contribute $\|\bar{\nu}_t\|_2$ to the sum in bin i, $1 \leq i \leq N_b$, to form the histogram h_t of N_b bins. Then h_t is normalized so that $\|h_t\|_1 = 1$. To preserve detailed temporal information, all the high energy histograms ($\|h_t\|_2 \geq \lambda$), are stacked as columns to obtain the histogram flow pattern (HFP) as $\mathbf{H}_f = [h_1, \cdots, h_{N_B}]$. Fig.2.1 summarizes all this procedure.

Block analysis of video, inherently acts as noise filter to reduce ill effects from

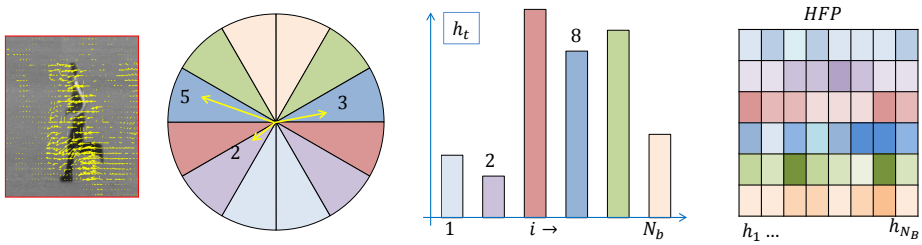


Figure 2.1: Construction of the histogram flow pattern (HFP)

rapid illumination changes and camera noise. Forming h_t according to primary angles, makes the representation to be independent of direction of motion (run left or right); and normalization makes it also scale-invariant. Gradient of the optical flow removes the locally constant camera motions [30], thus information about changes in the flow field is more robust and discriminative for activity analysis. Fig.2.2(a) shows the case of camera motion; the associated optical flow field [31] is occupied over entire video frame (see Fig.2.2(b)), which is erroneous and useless for HAR; however its gradient (see Fig.2.2(c)), preserves the salient motion information. First the derivative pattern for each b_t, is obtained like $\nabla\bar{\nu}_t = [\frac{\delta u}{\delta x}, \frac{\delta v}{\delta y}]^T$, then we follow the same procedure as explained previously to obtain the new histogram features $\mathbf{H}_{\nabla f}$.

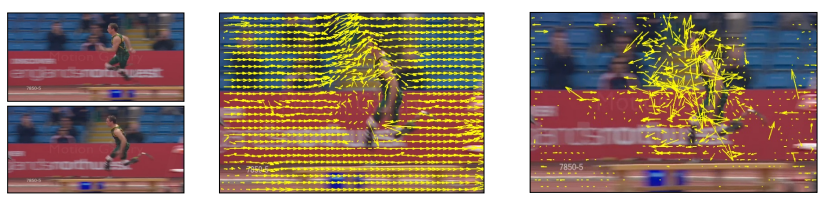


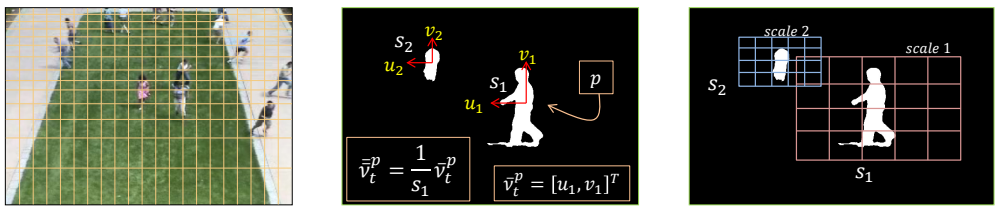
Figure 2.2: (a) Consecutive frames from moving camera video; (b) optical flow for these frames is occupied over the entire frame; (c) gradient of the flow captures the vital motion patterns necessary for underlying activity recognition

2.3.2 Freqencygrams

As alluded to the issues in sec.6.1, local descriptors capture only short movements within vicinity of the interest points, so they are limited in spatial and temporal scalability. Hence they are inadequate for describing activity with long-term motions. HFP circumvents this problem by preserving temporal information all long. However HFP is not useful in the raw form, because different activities are generally performed for different time duration. This renders the HFPs with different sizes for the same action class. The crucial information for HAR lies in temporal dynamics of the histograms. So to extract this information HFPs are analyzed in the frequency domain, with $\omega_k = \frac{2\pi k}{N_B}$ and $\omega_l = \frac{2\pi l}{N_b}$, like

$$\mathcal{H}_f(k,l) = \sum_{n=0}^{N_B-1} \sum_{m=0}^{N_b-1} e^{-i(\omega_k n + \omega_l m)} \cdot \mathbf{H}_f(n,m). \tag{2.1}$$

In reality, human activity consists of set of limb movements, which are changed gradually one after another and thereby giving rise to smooth HFP. These patterns are captured by low frequency components in $\mathcal{H}_f$. Abrupt motion changes correspond to the high frequency regions and are caused by illumination changes, camera shakes and background noise. Hence we preserve the only low frequency regions, which is nothing but performing a low pass filtering on original signal with the bandwidths ω_b and ω_B. This serves dual purpose by making features of fixed dimension and removing high frequency noise. Different actors may start performing the same action at different time instances. So to incorporate the shift invariance property, we consider the magnitude spectrum to construct the frequencygram feature. We are making use of low-level raw optical flow after cleaning it (via median flow and gradient flow mask) to extract meaningful information necessary for HAR. So the constructed Frequencygram is the mid-level



(a) (b) (c)

Figure 2.3: (a) Space bounding pattern (SBP) for areal view; (b) velocity normalization; (c) SBPs for frontal camera view at different spatial scales for building robust Spatiograms ($\widetilde{\mathcal{H}}_s$).

feature and it is given as follows,

$$\widetilde{\mathcal{H}}_f = \begin{cases} |\mathcal{H}_f(k,l)| & \text{if } k \in [-\omega_B, \omega_B] \text{ and } l \in [-\omega_b, \omega_b] \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

Similarly other feature $\widetilde{\mathcal{H}}_{\bigtriangledown f}$ is constructed. HFPs and frequencygrams for three different actions are depicted in Fig.2.7 and elaborated in sec.6.5.2. These features carry great temporal information, which flourishes more as the video sequence further unfolds itself frame by frame. For maneuvering HAR performance to high level, we make use of symbiotic relationship between all these histogram features via first approximating their probability distribution independently and then collectively inferring using them all.

2.3.3 Spatiograms

For scene understanding, to infer about what is going on in the video, the knowledge about where the things are happening also gives an essential clue. So it is important to emphasize on the distribution of spatial locations of active objects through the course of the event. Hence we build 2 dimensional histograms called 'spatiograms' ($\widetilde{\mathcal{H}}_s$), to keep account of spatial activity patterns. Between

identical objects, the one which is far from camera appear smaller and seems to move with lower velocity than the object closer to camera. To incorporate scale invariance in spatiograms, video is analyzed via different size spatial grids $G(i,j)$, called space bounding patterns (SBP), as shown in Fig.2.3(a). For aerial or fish eye views, scene background and camera information is used to lie SBPs for building spatiograms and for frontal camera view, SBPs with different scales are used (see Fig.2.3(c)). This scale for silhouette or blob $\mathcal{B}_i$ is proportional to its size s_i and that SBP remains stationary (for time τ) until silhouette leaves it; after this again a new SBP will be established. For making motion vectors scale invariant, they are also normalized to $\bar{\nu}_t^p = (1/s_i)\bar{\nu}_t^p$, where $\bar{\nu}_t^p$ is associated with pixel $p \in \mathcal{B}_i$ (Fig.2.3(b)). Foreground is given by, $\mathcal{F} = \underset{\forall i}{\cup}\mathcal{B}_i$. Each bin of the spatiogram is computed as,

$$\widetilde{\mathcal{H}}_s(i,j) = \gamma \sum_{\forall t \in \tau} \sum_{\forall p \in \{G(i,j) \cap \mathcal{F}\}} e^{-\alpha \|\bar{\nu}_t^p\|_2} \ , \tag{2.3}$$

where γ performs normalization to produce $\|\widetilde{\mathcal{H}}_s\|_1 = 1$. Constant α controls the feature quality and exponential term either captures how likely the active object in the scene remains stationary ($\alpha > 0$) or where the activity occurs ($\alpha < 0$).

2.4 High-Level Features: Abstracted Radon Profiles

2.4.1 Radon Transform

Radon transform (RT) [32] is widely used in tomography for creating an image from cross sectional scans of an object. In case of images, it is the projection of the image intensity along a radial line [32]. Let Radon profile (RP) be the collection of RTs of an image along various radial lines. If all consecutive radial lines are θ° apart, then it gives Symmetric RP (SRP_θ) otherwise asymmetric RP.

Asymmetric RP helps for extracting more information about the image along specific direction. As the number of RTs increases, the RP density rises. Let $f(x, y)$ be a 2-D function and $\rho = x\cos\theta + y\sin\theta$ be the parametric form of a line, where ρ is the smallest distance from the origin and θ is its angle with the x-axis. Let $\delta(\cdot)$ be the familiar Dirac delta function and $-\infty < \rho < \infty$, $0 \leq \theta < \pi$. Then the RT $\Re(\rho, \theta)$ of $f(x, y)$ can be given as

$$\Re(\rho, \theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\delta(\rho - x\cos\theta - y\sin\theta)dxdy. \tag{2.4}$$

In case of images, the RT reduces to just image rotation and intensity summation operations. Rotation of the input image corresponds to the translation of the RT along θ. Due to these computationally cheap operations we can easily construct dense RP for the human silhouette. Fig.2.4(a) shows a graphical illustration of RT at one particular angle for 2-D function $f(x, y)$. Original image can be reconstructed from RP by using Fourier Slice theorem [33]. It states that 1D Fourier transform of the projection $\Re(\rho, \theta)$ is equal to the 2D Fourier transform of an image evaluated on the line, where the projection was taken on. Consider $\hat{\Re}(x, y) = \int_{0}^{\pi} \Re(x\cos\theta + y\sin\theta, \theta)d\theta$ and $\Psi(u, v)$ be the 2D Fourier transform of $\hat{\Re}(x, y)$, then

$$f(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \sqrt{u^2 + v^2}\ \Psi(u, v)e^{j2\pi(ux+vy)}dudv. \tag{2.5}$$

Inverse RT is only used to assess the preservation of image information in RP. It is not used for feature formation. Fig. 2.4(b) shows the RP and the reconstructed artificial fingerprint image by inverse RT. In order to reconstruct original image perfectly, an infinite number of projections along all directions are required. Practically if the RP is dense enough for the sparse images like fingerprints then most of the original image information will be preserved.

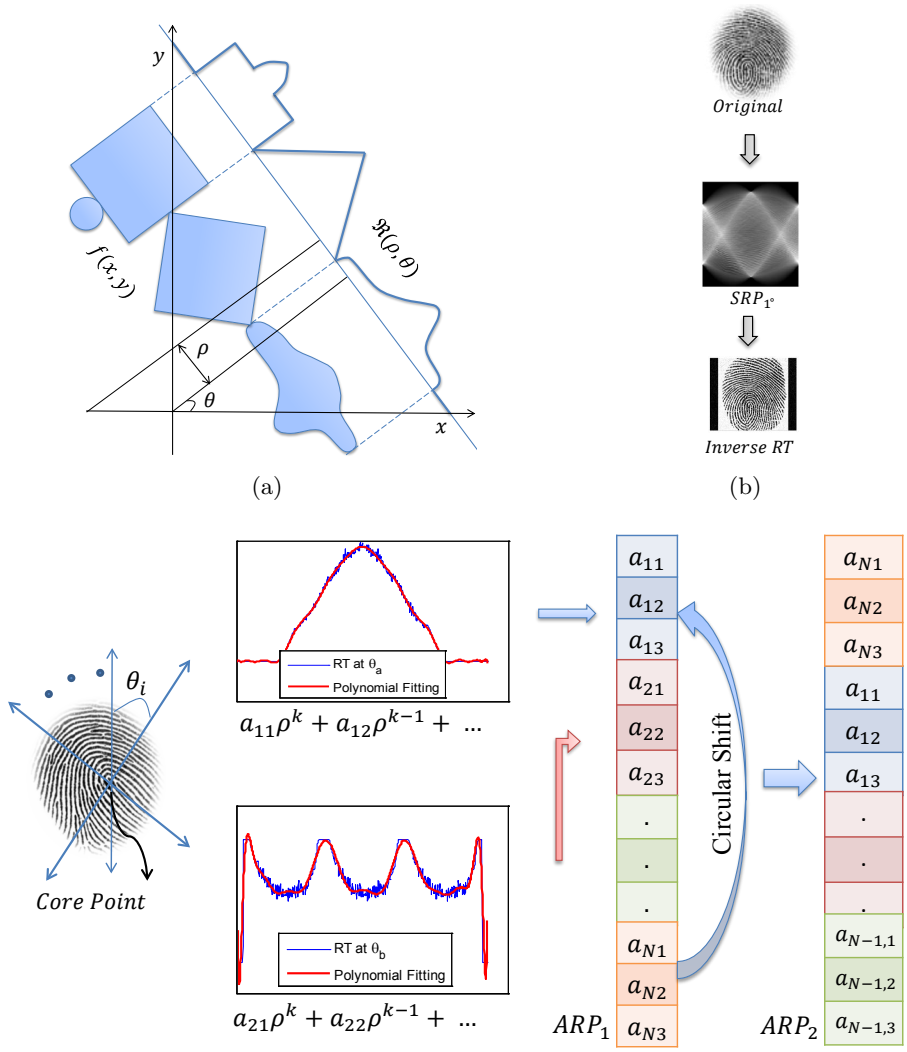


Figure 2.4: High-level features: Abstracted Radon Profiles (ARP). They are capable of capturing detailed oriented global information. This property is emphasized using fingerprint image, as it contains intricate spatial patterns. For HAR they are constructed using human silhouettes. (a) Illustration of the Radon Transform, (b) SRP and reconstruction of original image by inverse RT, (c) ARP feature construction and its property that ARP_2 is the ARP of rotated image.

2.4.2 Abstracted Radon Profiles (ARP)

Let $w(x, y)$ be the zero mean white noise then an image can be represented as $\hat{f}(x, y) = f(x, y) + w(x, y)$. After taking its RT and using distributive property [32], $\Re(w(x, y))$ is line integral of the noise, which is constant for all directions and is equal to the mean value of the noise. Thus zero mean white noise does not contaminate the features based on RT.

It is computationally impractical to store entire dense RP for each video frame. This profile is abstracted to a vector by using polynomial fitting. Each RT from the RP is approximated with higher order polynomial of ρ_i for each θ_i as

$$a_i^k \rho_i^k + a_i^{k-1} \rho_i^{k-1} + \cdots + a_i^1 \rho_i + a_i^0 \approx \Re(\rho_i, \theta_i). \tag{2.6}$$

Increasing the value of k, increases the computational effort with only small increase in performance, as elaborated in our work [28]. Furthermore, only few polynomial coefficients per RT need to be captured. This will help to store most of image information from dense RP in the form of feature vector as

$$\varphi^T = \langle (a_0^k, \ldots, a_0^{k-m}), \ldots, (a_i^k, \ldots, a_i^{k-m}), \ldots, (a_N^k, \ldots, a_N^{k-m}) \rangle. \tag{2.7}$$

Here RP constitutes $N + 1$ different RTs and top order $m + 1$ coefficients are taken after fitting the k^{th} order polynomial to each transform. Parameters SRP_θ, k and m control the ARP feature properties. By changing these parameters, different ARP features are obtained for the same image, which capture different amount of information and hence show varied performance. Fig. 2.4(c) illustrates this feature formation procedure briefly. In case of SRP_θ, when we perform circular shift to ARP feature φ by $M(m + 1)$ amount with M is an integer, we will get the feature vector for $M\theta^\circ$ rotated image. Thus randomly oriented test image can be analyzed in the similar way as trying their all possible orientations for alignment but without actually doing it. Hence this feature serves more computationally efficient way than the brute-force method for image alignments.

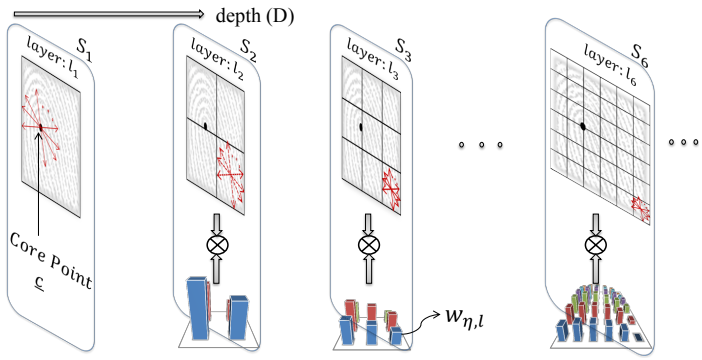


Figure 2.5: Multilayer architecture: A independent patch analysis framework to collect spatially local as well as global image information using the ARP features.

2.4.3 Multilayer Architecture for ARP Features

RT with the center of the image as the origin of ρ-θ coordinate system is translation invariant. In case of fingerprints, we consider core point and in HAR the silhouette median as the reference point. Core point is the singular point in a fingerprint that exhibits the maximum ridge line curvature. It is found by Poincaré index method [34]. In case of extremely degraded images, center of the segmented fingerprint is considered as the reference point. RT is not scale invariant but alignment of the training images can help RT to achieve scale invariance.

For taking local characteristics of fingerprint into account, images are analyzed in independent multiple layers. Progressive image patching is done in each layer as shown in Fig. 2.5. Images are analyzed via l^2 patches in layer l. Each patch $\eta \in [1, \ldots, l^2]$ in layer $l \in [1, \ldots, L]$ analyzed independently for HAR by action silhouette matching. As the total number of layers L increases, patch size gets smaller and smaller and more fine local details of the image are captured. There is tradeoff between space-time complexity and the performance. One way to reduce computational effort is by constructing RP with the density proportional to the patch size. Because information of small patch can be captured

by few RTs. Thus smaller the patch, sparse should be the RP. During testing, image is passed through each layer and every patch produces its recognition score (e.g. likelihood, sparse reconstruction score [28]) $s_{\eta,l}$. Intuitively patches nearby core point (median point) should contain more fingerprint (silhouette) information according to definition of core point. Consider $w_{\eta,l}$ be the weight associated with each patch. If the patch η_i is closer to core point than patch η_j then we set $w_{\eta_i,l} > w_{\eta_j,l}$. One simple and static way to do this is to make $w_{\eta,l} \propto \mathcal{D}^{-1}(\underline{c}, \underline{\eta})$, where $\mathcal{D}(\underline{c}, \underline{\eta})$ is the Euclidean distance between core point $\underline{c}$ and the patch center $\underline{\eta}$. Score in each layer is combined as

$$S_l = \sum_{\eta=1}^{l^2} w_{\eta,l} \cdot s_{\eta,l} \quad \text{such that} \quad \sum_{\eta=1}^{l^2} w_{\eta,l} = 1. \tag{2.8}$$

The score S_1 gives more information about the global features than $S_{l>1}$. On the other hand $S_{l>1}$ represents more local properties of video frame (fingerprint) than S_1. Oriented features of the silhouettes are emphasized by RP of the entire image, so $S_1 \in [0, 1]$ is more important evaluation criterion than other scores. Thus when S_1 is at either of its range extremities, recognition decision can easily be done. But the ambiguity arises when S_1 falls in middle region of its range and then we cannot rely only on it. Thus we need to take help from multiple layers depending upon the unreliability amount $\left(\alpha = \frac{1-2\cdot|S_1-0.5|}{1+2\cdot|S_1-0.5|} \in [0, 1]\right)$ of S_1; which is the non-linear function of S_1, which produces high value when S_1 lies in the middle region of its range, as discussed earlier.

2.5 Inference and Classification

Training features are constructed as $\boldsymbol{x} = [\widetilde{\mathcal{H}}(i, j), i, j]^T$. Then Gaussian mixture model is used to represent the feature distribution as a weighted sum of K Gaussian components like, $p(\boldsymbol{x}|\Theta_c^q) = \sum_{m=1}^{K} \beta_m^{q,c}\, p(\boldsymbol{x}|\theta_m^{q,c})$, where Θ_c^q is the mix-

(a)

Method	Accuracy (%)
Barattini et al. [3]	78.0
Bomma et al. [36]	98.0
Our method	98.3

(b)

(c)

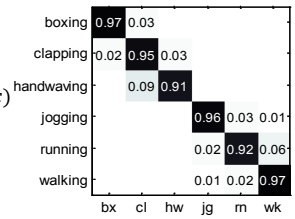


Figure 2.6: (a) Sample frames from all the datasets: UMN crowd behavior (1^{st} row), KTH (2^{nd} row), Weizmann (3^{rd} row), HWU gesture (4^{th} row). It also shows the viewpoint and illumination changes; (b) performance comparison with [3] and [36] for HWU hand gesture dataset; (c) per class HAR rate (confusion matrix) of the proposed method on KTH dataset.

ture model corresponding to the class c and feature type $q \in \{\widetilde{\mathcal{H}}_f, \widetilde{\mathcal{H}}_{\triangledown f}, \widetilde{\mathcal{H}}_s\}$; $\beta_m^{q,c}$ is mixing weight of the m^{th} component and density of each component is the normal probability distribution, $p(\boldsymbol{x}|\theta_m^{q,c}) = \mathcal{N}(\boldsymbol{\mu}_m^{q,c}, \boldsymbol{\Sigma}_m^{q,c})$. Parameters $(\beta, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ are iteratively estimated during training using Expectation Maximization (EM) algorithm [35]. Given a test video with features $\boldsymbol{X}_q = \{\boldsymbol{x}_q^i\}_{i=1}^n$, (e.g. $\boldsymbol{x}_q^1 = [\widetilde{\mathcal{H}}_q(1,1), 1, 1]^T$), then its log-likelihood corresponding to class c is given by

$$L(\boldsymbol{X}; \Theta_c) = \sum_{\forall q} \sum_{i=1}^{n} \log \sum_{m=1}^{K} \beta_m^{q,c}\, p(\boldsymbol{x}_q^i|\theta_m^{q,c}). \tag{2.9}$$

In multiclass classification, test sample is assigned to the class (c^*) which maximizes (2.9) as,

$$c^* = \underset{c}{\operatorname{argmax}}\, L(\boldsymbol{X}; \Theta_c). \tag{2.10}$$

2.6 Experimental Results

2.6.1 Datasets and Evaluation Details

We evaluate the proposed approach on publicly available four benchmark datasets, viz. KTH action [6], HWU gesture [3], Weizmann action [4] and the UMN crowd (abnormal activities) dataset [5]. Each of these consists of different types of

human activities, which are delineated in Fig.2.6(c) and Fig.2.8 by action or gesture labels of the corresponding confusion matrix. Sample frames from all these datasets are shown in Fig.2.6(a), which shows a wide range of variations in viewpoints, background noise and illumination changes. Actions in KTH dataset were performed by 25 subjects in 4 different scenarios (indoor, outdoor, scale variation, different cloths), with variation in the execution speed of action and illumination changes. Weizmann dataset contains 90 low-resolution videos with 10 natural actions performed by 9 different subjects. HWU dataset consists of 10 gestures performed multiple times by 5 actors. UMN is the crowd activity dataset which consists of 3 different scenes of crowd escape events having total 7740 frames with 320×240 resolution.

We fix the parameters as, $N_F = 3$, $N_b = 32$, $\lambda = 0.1$, $\omega_b = 2\pi$, $\omega_B = 0.5\pi$ radians/sample; and follow the same experimental setup as described by the dataset publishers. Implementation is done in Matlab with Intel Core i3 2.4 GHz processor and 2 GB RAM. For UMN dataset, we train the model using only normal activities from one scene and test on all remaining activities from all the scenes. For KTH dataset, we used the standard splitting, dividing samples into training and testing set (16 vs 9 subjects). And for the rest, we applied leave-one-out cross validation as others have used in the research community.

2.6.2 Performance Analysis

Before diving into detailed empirical performance analysis, we present the compactness and discriminability characteristics of the frequencygrams ($\widetilde{\mathcal{H}}_f$s). We considered two fairly related actions (Fig.2.7(a) *walking*, (b) *running*) and one relatively distinct action ((c) *handwaving*) from the KTH dataset. Fig.2.7 shows the sample frames (1^{st} column), HFPs (2^{nd} column) and $\widetilde{\mathcal{H}}_f$s ($3^{rd}$ column) of these actions. These actions were performed at different rate and for different

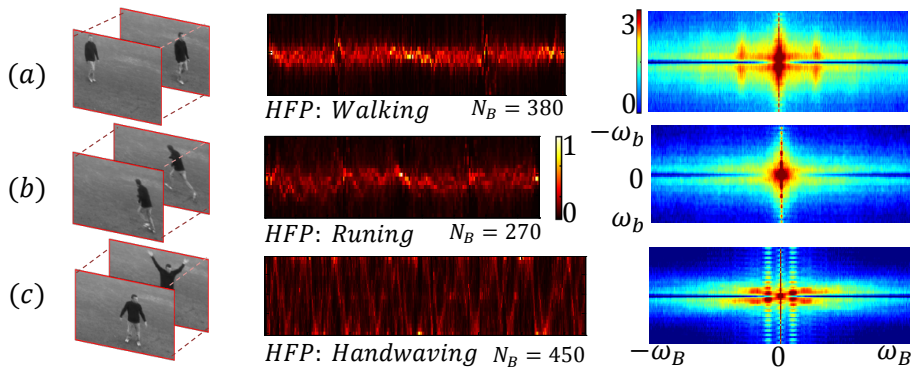


Figure 2.7: HFPs and the corresponding Frequencygrams ($\widetilde{\mathcal{H}}_f$)

time duration (N_B), thereby producing HFPs with different sizes. But corresponding all $\widetilde{\mathcal{H}}_f$s have an identical size defined by the specified bandwidths (ω_b and ω_B), which makes the compact and equi-dimensional representation of action sequences. HFPs of *walking* and *running* are quite similar but the corresponding $\widetilde{\mathcal{H}}_f$s are able to uncover the underlying hidden motion dynamics and thereby amplifying the discriminability between the action sequences. In *handwaving*, the motion is distributed over all directions, so different bins of h_t contribute at different time. Thus in overall, there is variation within each h_t as well as among different h_ts. This information is highlighted in $\widetilde{\mathcal{H}}_f$ by high frequency components along both vertical (b-axis) and horizontal (B-axis) direction. Whereas, *walking* and *running* has mainly horizontal global motion pattern, so their HFPs are concentrated within certain fixed bins of h_t. Hence corresponding $\widetilde{\mathcal{H}}_f$s consist of high frequency components only along B-axis. It is interesting to observe that, $\widetilde{\mathcal{H}}_f$ of *walking* contains higher frequency components (along B-axis) than $\widetilde{\mathcal{H}}_f$ of *running*. It may be due to the fact that, *walking* consists of different set of limb movements and as the action is being performed slowly, the motion vectors capture all the characteristic movements (gait of a person) and causing h_ts to vary rapidly along temporal direction. Whereas, in case of *running*, the action is

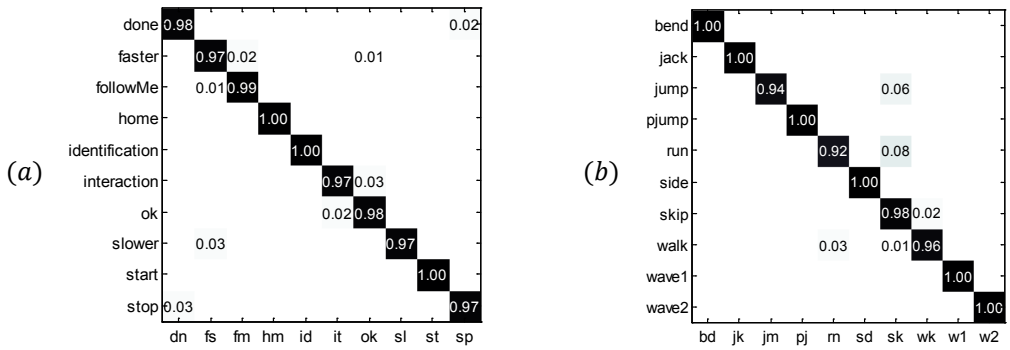


Figure 2.8: Recognition results for each activity class using proposed method. (a) confusion matrix for HWU hand gesture [3] dataset; (b) confusion matrix for Weizmann [4] dataset.

performed so rapidly that small motion vectors associated with limb movements are dominated by most of the other motion vectors representing only horizontal global movement of the person.

Computationally $\widetilde{\mathcal{H}}_s$, $\widetilde{\mathcal{H}}_f$ and $\widetilde{\mathcal{H}}_{\triangledown f}$ run at 31, 26 and 21 frames/sec respectively, which is well suited for the real-time applications. The detailed HAR analysis for each activity on three benchmark datasets is given in Fig.2.8 and 2.6(c), via confusion matrices. On KTH dataset, the proposed method produces 94.67% mean average precision (mAP) over all classes, where it slightly interchangeably con-

Table 2.1: Average HAR accuracy (%) comparison on benchmark datasets

Activity recognition method (Year)	KTH [6]	Weizmann [4]
Wang et al. (2009) [10]	89.0	97.8
Kovashka et al. (2010) [37]	94.5	-
Campos et al. (2011) [38]	91.5	96.7
Xinxiao Wu et al. (2011) [13]	94.5	-
Gao et al. (2012) [39]	92.0	-
Luo and Hu (2013) [17]	-	96.7
Jiaxin Cai et al. (2013) [27]	-	90.0
Our method	**94.7**	**98.0**

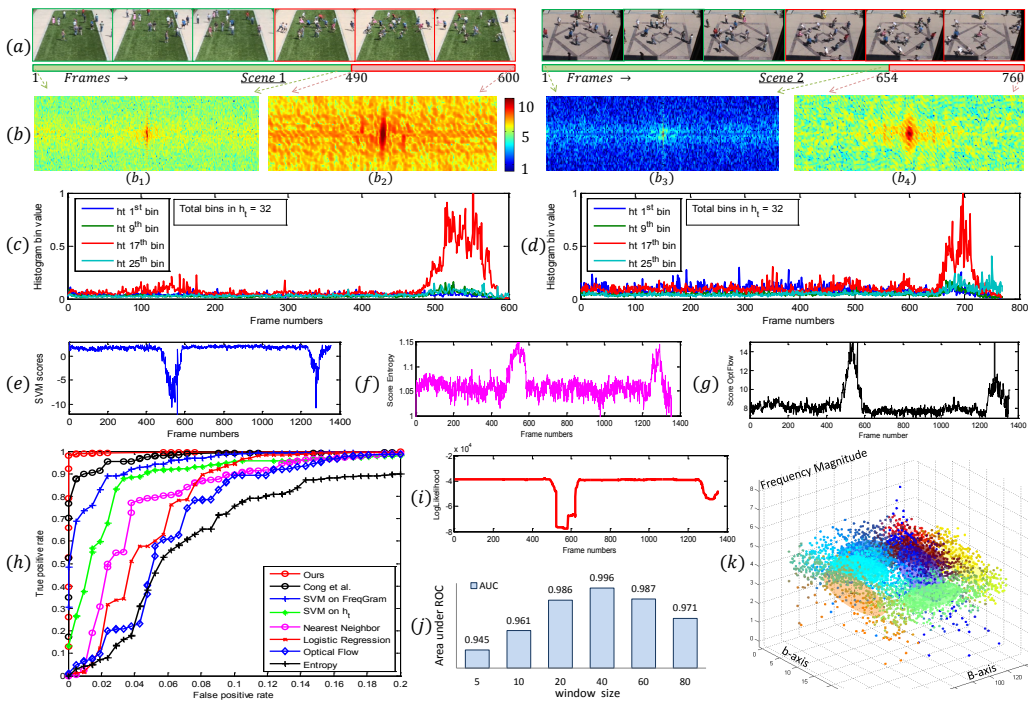


Figure 2.9: Detailed experimental analysis for two activities from UMN dataset.

fuses for *clapping*↔*handwaving* and *running*↔*walking* activity pairs (Fig.2.6(c)). For gesture recognition on HWU dataset, it achieves 98.3% mAP (see confusion matrix in Fig.2.8(a) and comparison with others in Fig.2.6(b)) and on Weizmann dataset it produces 98% mAP. The diagonal nature of the confusion matrices (Fig.2.8) depicts discriminability of the proposed features. Table 2.1 compares average HAR rate of the state-of-the-art methods on KTH [6] and Weizmann [4] human activity datasets.

2.6.3 Abnormal Activity Detection

The proposed framework is more general and can be used for analyzing wide range of human activities. It is tested here for abnormal activity detection on UMN dataset [5]. Fig.2.9 gives the comprehensive experimental results on UMN

dataset for analyzing crowd activities. Here we make use of only Frequencygrams, the mid-level features. Other mid-level feature Spatiogram is tested for action and gesture recognition, whereas high-level ARP features are extensively tested for fingerprint recognition in our work [28]. Part **(*a*)** shows few sample frames from two testing videos (*Scene* 1 and 2). For both cases, first three frames (green border) belong to normal event and last three (red border) belong to abnormal event. The bars below them show the frame-by-frame ground truth for both videos, where normal events (green colorbar) are followed by abnormal ones (red colorbar). Part **(*b*)** shows the different frequencygrams for different segments from these scenes. The frequencygram shown in (b_1) is constructed from the HFP belonging to only normal events (green colorbar in (a)) from scene 1, and (b_2) is for abnormal events from the same scene. Similarly frequencygrams (b_3) and (b_4) are for normal and abnormal events from scene 2 respectively. It shows that HFP follow completely different dynamics in normal and abnormal scenarios. For abnormal events, there is a great intra-h_t as well as inter-h_t variability, which causes high frequency components to appear in (b_2) and (b_4) across both b-axis and B-axis. Part **(*c*)** and **(*d*)** show the inter-h_t variations of some of its bins for both scenes. Bin 17 (red) responds actively to the abnormal events in both scenes but it produces many false positives (see frames 100 to 200 in scene 1; and several occasions in scene 2). On the other hand in scene 2, its response died earlier (around frame 720) than actual termination of the abnormal events, where only bin 25 (cyan) responded correctly. This shows that each bin carry different information of the underlying events, so it is important to analyze both intra and inter-h_t dynamics for proper scene understanding, that is what frequencygrams do. For testing, both scenes are concatenated to get sequence of 1360 frames. Part **(*e*)** shows SVM decision scores for 2 class (normal, abnormal events) classification. For training SVM, both normal and abnormal event samples were used

with h_ts as features. Similarly parts **(f)**, **(g)** and **(i)** show decision scores for classification using entropy, row optical flow and the proposed framework. Disordered and abrupt motion of subjects constitutes abnormality, so entropy and optical flow rises as expected during those events. But this rise is not sufficient to achieve good discrimination between normal and abnormal events. Where as seen in (i), the proposed method decision score (given by (2.9)), shows great discriminability (see y-axis scale is in the order of 10^4). In addition to this, it is stable whereas the other scores show much noisy variations even during normal events. Part **(h)** shows the frame level ROCs. The proposed method outperforms Cong et al. [40], baseline SVM and other pattern classification techniques. SVM using frequencygrams show greater accuracy than mere histograms (h_ts) features, which again shows their importance. For training, the proposed method uses samples from only normal events to estimate the parameters of GMM model and during testing the event which show lower likelihood are classified as abnormal ones. Whereas SVM requires training data from both classes. All these testing is performed online by employing sliding window protocol i.e. HFP is analyzed in small consecutive overlapping subsets. Part **(j)** shows the effect of window size on classification accuracy. Smaller window size is unable to capture enough temporal dynamics, where as much larger size (resembles batch processing) introduces lag between ground truth and predicted labels, so both of them result in loss in performance. Part **(k)** shows the trained Gaussian mixture model that we have used in these experiments, where Gaussian 8 components sufficed to capture whole data distribution adequately. The overall abnormality detection performance comparison on UMN dataset is shown in Table 2.2. Hence high recognition rate for wide range of activities (like human action, gesture and crowd behavior) along with low computational cost of features, shows the effectiveness and generality of the proposed activity recognition framework.

Table 2.2: Performance comparison using area under receiver operating characteristic curve (AUC) measure with state-of-the-art methods for abnormal event detection in UMN dataset.

Method	AUC
Optical Flow	0.84
Nearest neighbor learning	0.93
Mehran et al. (Social force model) [41]	0.96
Wu et al. (Chaotic invariant analysis) [42]	0.99
Cong et al. (Sparse reconstruction cost) [40]	0.975
Sandhan et al. (Low level motion features) [18]	0.981
Saligrama and Chen (Local stats aggregates) [43]	0.985
Ours (Dynamics of the motion histograms: mid-level features)	**0.996**

2.7 Conclusions

We present a novel approach for activity recognition by proposing new mid-level (*Frequencygrams*, *Spatiograms*), high-level (*Abstracted Radon Profiles*) features and making inference by maximizing collective (over all features) likelihood of the test sample. Spatiograms capture the distribution of spatial location of the activity occurring over the course of time. Frequencygrams extract implicit spatio-temporal knowledge by uncovering the hidden dynamics of the motion histograms by analyzing them in frequency domain. ARP captures global oriented image (action silhouette) information. Using proposed multilayer architecture for ARP, we can also extract local image information. They provide a natural, compact and discriminative representation for reciprocating actions. They can represent detailed temporal information of the activity, unlike other methods e.g. bag-of-features which ignores the spatio-temporal dependencies. They are fast and easy to compute, so well suitable for the real-time applications. The proposed method has an advantage of being robust to camera motions and it does not necessitate extra video preprocessing overload unlike other existing features.

Chapter 3

Low-Level Features and Proximity Clustering

3.1 Introduction and Prior Work

In the realm of automatic surveillance, some of the methods cluster a set of low-level motion features into trajectories [44] and some try to identify each member in the scene based on motion cues [45]. These methods are based on human motion recognition and usually they necessitate the segmentation and tracking of each person. It requires high computational cost, in addition the recognition accuracy depends on the accuracy of video preprocessing and tracking methods [46], [47]. Tracking-based anomaly detection algorithms [48], [49] consider object trajectory as the key feature and model the object movement based on previously clustered trajectories. Hidden Markov model (HMM) based representation of trajectories has the problem of overfitting due to data shortage, so the HMM representation based on several similar samples acquired by dynamic hierarchical clustering [50] has been used to perform supervised learning of normal trajectory

patterns. These *object based* approaches treat crowd as a collection of individuals. These methods face several challenges such as the severe object occlusion in crowded scene, the exploration of multimodality video sensor and multimodal data fusion [51] and the poor quality of the surveillance footage.

On the other hand, *holistic* approaches treat the crowd as a single entity, without using object tracking or segmenting individual entities. The crowdness indicator [52] helps to distinguish static from moving crowds and can be used to find the potential danger [51]. Like *object based* approaches , HMM has also been used in *holistic* approaches to characterize normal behavior of a crowd [53]. The only difference between them is that the former use object detection and tracking methods to model trajectories, while later use an optical flow to model the entire crowd. Mixtures of dynamic textures are useful for modeling normal crowd behavior [54], where outliers are treated as anomalies. Coherent and incoherent crowd scenes can be represented at the same time using combination of Langrangian particle dynamics approach together with chaotic modeling [55]. These approaches are only useful for management of high density crowd in confined spaces. But their performance degrades in case of sparse scenes.

Some methods based on foreground blob analysis, binary silhouettes and Radon transform are useful for human activity recognition [32], [28], [56], but they are computationally expensive for real time video analysis. So we propose efficient motion features which can be computed easily from motion vectors. We also try to combine both, object based and holistic approaches for crowd analysis. This helps to handle the wide range of scenarios from sparse to dense crowd.

Motivation: Fundamentally it is hard to answer, what is an abnormal event. Completely normal event in one scenario might turn out to be abnormal in the another situation. For instance, consider the events like, 'a cattle is grazing in a large open field' and 'a cattle is grazing in a house backyard'. In video

processing where the background get subtracted, both are the same events; except the scenario where the action of grazing is being performed. An important thing that separates both events apart is the usual frequency of their occurrences. The grazing is common action and happens frequently in an open grass field unlike the house backyard, which makes it normal event for that situation. When the grazing is repeatedly performed many times in the house backyard (probably in suburb) then it becomes usual activity and should be classified as the normal event. Hence anomalous events occur relatively infrequently.

Most of the time surveillance cameras are capturing the normal events. Now when we look at the surveillance video in the feature space then we would find various clusters of feature vectors. Those clusters might be spatially well separated but semantically they would all alike with the reasoning that they will be representing the normal events. The outliers in that feature space are nothing but the unusual or abnormal events. Being unaware of the number of clusters or their properties like spread or density, we can not use the conventional clustering techniques. When operated with two clusters, each for normal and abnormal events, the methods like k-means (KMC), hierarchical (Hie) and fuzzy c-means (FCM) clustering perform very poorly. Spectral clustering (Spc) [57] and density based clustering techniques (DBS) [58] fail to improve performance due to their high sensitivity to affinity matrix [59] and algorithm parameters [60] respectively. So we propose a novel proximity based clustering algorithm for abnormality detection. Feature space is constructed from the proposed motion features.

We will first introduce the proposed motion features in sec. 3.2 and then proximity clustering algorithm in sec. 3.3. In sec. 3.4 evaluation of the proposed method over an artificial database, their comparison with existing clustering methods and its validation over real surveillance video database for abnormality detection are presented. We conclude our work in sec. 3.5.

3.2 Motion Features

The video frame at time t, is analyzed via N_{mn} macro-blocks. Each macro-block, $\mathcal{M}(i,j)$ is of size $h \times w$, and a dominant motion vector $V_{i,j}^t = [V_{x_{i,j}}^t, V_{y_{i,j}}^t]$ is associated with it. If v_p^t is a motion vector associated with pixel p then

$$V_{i,j}^t = \frac{1}{h \times w} \sum_{p \in \mathcal{M}(i,j)} v_p^t.$$

Different features use different block size to extract diverse information from the motion vectors.

- **Circulation** (f_c): The collection of all motion vectors v_p^t, constitutes the motion vector field (MVF). f_c measures the amount of local spin around the perpendicular axis to the MVF. For constant or zero MVF, the f_c is zero.

$$f_c\big(\mathcal{M}(i,j),t\big) = \nabla \times V_{i,j}^t = \Big(\frac{\partial V_{y_{i,j}}^t}{\partial x} - \frac{\partial V_{x_{i,j}}^t}{\partial y}\Big). \tag{3.1}$$

It helps to capture the turbulence in MVF caused by sudden abrupt movements of the subjects in the video. It also measures the rigidity and tries to highlight the dynamics of MVF.

- **Motion Homogeneity** (f_h): The deviation of the dominant f_c from the global circulation within the frame, indicates the motion homogeneity of the event. The region of dominant circulation is given as,

$$\mathbb{R}_{f_c}(t) = \underset{\mathcal{M}(i,j)}{\operatorname{argmax}}\, f_c(\mathcal{M}(i,j),t), \tag{3.2}$$

then,

$$f_h(t) = \frac{exp\big(\big|\nabla \times [\frac{1}{h \times w} \sum_{p \in \mathbb{R}_{f_c}(t)} v_p^t]\big|\big)}{1 + exp(|f_c(\mathcal{M}(i,j),t)|)}. \tag{3.3}$$

• **Motion Orientation** (f_o): It captures the directional information of the scene. The range of motion orientation is quantized into N_0 bins, each of size $\frac{2\pi}{N_0}$. Each $\mathcal{M}(i,j)$ is associated with orientation of its $V^t_{i,j}$ as, $\Phi^t_{i,j} = arctan(V^t_{y_{i,j}}, V^t_{x_{i,j}})$. Let $\mathcal{K}$ and $\mathcal{I}$ be the N_0 dimensional vectors defined as, $\mathcal{K} = [1, 2, \cdots, N_0]^T$ and $\mathcal{I} = [1, 1, \cdots, 1]^T$. Consider a vector valued Heaviside step function as follows,

$$H(\mathbf{x}) = [u(x_1), u(x_2), \cdots, u(x_{N_0})]^T,$$

where,

$$u(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise,} \end{cases}$$

then the feature vector f_o is given as,

$$f_o(t) = \frac{1}{N_{mn}} \sum_{\forall \mathcal{M}(i,j)} H\Big(\frac{\pi}{N_0}\mathcal{I} - \big|\Phi^t_{i,j}\mathcal{I} - \frac{2\pi}{N_0}\mathcal{K}\big|\Big). \tag{3.4}$$

• **Stationarity** (f_s): It captures how likely the active objects in the scene remains stationary. In the video, objects are tracked [61] to get the set of bounding boxes $\mathbb{B}$. Let α be the constant tuning parameter for adjusting the feature quality. Stationarity map S_{map} is found first to obtain f_s as follows,

$$S_{map}(p,t) = \begin{cases} exp(-\alpha|v^t_p|) & \text{if } p \in \mathbb{B} \\ -1 & \text{otherwise.} \end{cases}$$

$$f_s\big(\mathcal{M}(i,j), t\big) = \frac{1}{h \times w} \sum_{p \in \mathcal{M}(i,j)} S_{map}(p,t). \tag{3.5}$$

• **Temporal Smoothing** ($f \rightarrow f^{\tau}$): Sudden illumination changes and camera shakes make frame by frame feature extraction susceptible to errors. In addition, the change in motion vectors within few consecutive frames is very small. So the above features (f) are averaged temporally over N_τ consecutive frames. Let t_τ be any key frame then next key frame is given as $t_{\tau+1} = t_\tau + N_\tau$ and final features are obtained as,

$$f^{\tau}(t_\tau) = \frac{1}{N_\tau} \sum_{t \in (t_\tau, t_{\tau+1})} f(t). \tag{3.6}$$

So we are obtaining the features only from the key frames. For experimentation (sec. 6.5), N_τ is set 20 and within that time there is not any significant change in the events. So this video analysis via small video clips, helps to suppress noise from sudden illumination changes and failure of object tracking method.

3.3 Algorithmic Framework

3.3.1 Discussion

It is assumed that the database X (the set of m dimensional input samples) contains an unique dominant pattern. Here, X is comprised of various feature vectors (x_i), which are extracting different information from the given surveillance scenario. For the dominant one-class case like normal events occurring in the video, the various x_i will try to convey the same information. So the elements of X may get scattered in m dimensional feature space at different places but they will be indicating the same underlying pattern. When the actual event in the given problem gets repeated several times, the feature pattern will also start repeating. This leads to formation of the many clusters, which are scattered in that feature space. The thing is that, we do not know how many clusters

there will be, their location (cluster centers), density or spread. They might be of irregular shapes like interwind together or having spiral shapes with leaving voids inbetween them. The only thing we know here is that all of the clusters would be representing the same phenomenon. So we can say that there is only one cluster which is poriferous in nature.

Popular clustering techniques like k-means, hierarchical and fuzzy c-means clustering perform very poorly in this case. These types of clustering methods are restricted to the data having notion of a centroid. They can not handle the non-globular data of different sizes and densities. It also fails to identify outliers and the final clustering result depends upon the initial choice of seeds.

The density based clustering technique like DBScan (DBS) can find irregular shaped poriferous clusters. Because unlike centroid based clustering techniques, the cluster structure is represented by certain important data points themselves. It finds all these separated clusters independently and return their indices, which can be considered as same index to get the unique porous type of cluster. But the DBS depends upon the parameters like maximum neighborhood radius (ϵ) and the minimum number of points in ϵ neighborhood of the core point. And it is very sensitive to these parameters [60]. That is the slight change in ϵ, widely changes the output cluster structure. Also unlike FCM clustering, the cluster structure is rigid in nature i.e. each point either belongs to cluster or not. There is no 'degree of belongingness' assigned to each point, which leads to degradation of the pattern classification performance with increase in false alarms. When the different clusters have very different densities or they are in hierarchies, then being dependent on sensitive parameters DBS fails to find proper cluster structure. Also when the underlying problem involves heavy computational tasks like video processing, then speed is the major concern while applying density based clustering methods.

So to solve these problems, a new Proximity clustering algorithm has been proposed here. It is useful for capturing an imperative pattern, which might be spatially diverse but semantically similar. It represents the porous cluster in the form of flexible structure using proximity vectors along with degree of belongingness is assigned to each of them.

3.3.2 Algorithm

Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ be the dataset of m dimensional N input samples. Then $\Delta X = \{\mathbb{D}(\mathbf{x}_i, \mathbf{x}_j) | 1 \leq j < i \leq N\}$ is multiset of cardinality, $|\Delta X| = \binom{N}{2}$, where $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \cdots, x_{i,m}]^T$ and

$$
\begin{aligned}
cov(\mathbf{x}_i, \mathbf{x}_j) &= \frac{1}{m} \sum_{l=1}^{m} (x_{i,l} - \bar{x}_i)(x_{j,l} - \bar{x}_j) \\
std(\mathbf{x}_i) &= \sqrt{\frac{1}{m} \sum_{l=1}^{m} (x_{i,l} - \bar{x}_i)^2}\,, \quad \bar{x}_i = \frac{1}{m} \sum_{l=1}^{m} x_{i,l} \\
\mathbb{D}(\mathbf{x}_i, \mathbf{x}_j) &= 1 - \Big| \frac{cov(\mathbf{x}_i, \mathbf{x}_j)}{std(\mathbf{x}_i) \cdot std(\mathbf{x}_j)} \Big|.
\end{aligned} \tag{3.7}
$$

Here $\mathbb{D}(\cdot)$ is treated as the distance measure between two feature vectors. The multiset, ΔX contains all the information necessary to preform proximity clustering. The distance of k^{th} nearest neighbor give an idea about the density of the surrounding points. Let $d^{(k)}(\mathbf{x}_i)$ be the k^{th} minimum distance between $\mathbf{x}_i$ and rest of the elements from X, given as

$$
d^{(k)}(\mathbf{x}_i) = \underset{\forall \mathbf{x}_j \in X \setminus \mathbf{x}_i}{\mathrm{k^{th}min}} \mathbb{D}(\mathbf{x}_i, \mathbf{x}_j). \tag{3.8}
$$

Note that, $d^{(k)}(\mathbf{x}_i) \in \Delta X$, so it can be found readily. Let $(X, d^{(k)})$ be a metric space and let the elements of X are classified, as degree to which they belong to a unique cluster, by the function $f : X \to [0, 1]$. As the large porous cluster consists of small scattered clusters, $\mathbf{x}_j$ shares approximately similar class as $\mathbf{x}_i$

if $\mathbf{x}_j$ is the nearest neighbor to $\mathbf{x}_i$, i.e. $f(\mathbf{x}_i) \approx f(\mathbf{x}_j)$. Metric space with $d^{(k)}$, captures this property and changing k, changes the corresponding metric space.

Here we clarify few more notations to simplify the explanation of the algorithm. Let $\mathbb{S}$ be the ordered set of elements and Ⱨ be the any random index set with $|\text{Ⱨ}| \leq |\mathbb{S}|$, then $\delta(\mathbb{S}, \text{Ⱨ})$ is the set of elements picked from $\mathbb{S}$ according to indices in Ⱨ. Thus $|\delta(\mathbb{S}, \text{Ⱨ})| = |\text{Ⱨ}|$. Let $\mathbb{I} = \{1, 2, \cdots, N\}$ be the index set corresponding to the elements of X. Now define the function $construct(\mathbb{S}_1, \mathbb{S}_2)$ which combines any two sets of equal cardinality (i.e. $|\mathbb{S}_1| = |\mathbb{S}_2|$), to form a pair-set. For example, $construct(d^{(k)}(X), \mathbb{I}) = \big\{(d^{(k)}(\mathbf{x}_1), 1), (d^{(k)}(\mathbf{x}_2), 2), \cdots, (d^{(k)}(\mathbf{x}_N), N)\big\}$. These pair-sets are nothing but the different 'layers' of neighborliness measure around feature points. The relationship among the elements of the pair-set gives an information about the distribution of the entire data and thereby about the unique porous cluster. For example, initial layer is given as, $\mathbb{L}_1 = construct(d^{(k)}(X), \mathbb{I})$. Whereas the function $distruct(\cdot)$, separates the pair-set into its two original sets. The function $sortpairset(\cdot)$ rearranges the pair-set elements in ascending order, according to only 1^{st} element ($d^{(k)}(\cdot)$ for $\mathbb{L}_1$) of the pairs. Forward difference operator $\Delta^{(+)}(\cdot)$ finds the difference between adjacent elements of an ordered set, except for the 1^{st} element, which is subtracted from itself. For example, $|\Delta^{(+)}(\mathbf{x}_i)| = \big\{|x_{i,1} - x_{i,1}|, |x_{i,2} - x_{i,1}|, \cdots, |x_{i,m} - x_{i,m-1}|\big\}$.

Proximity clustering is descrived in algorithm 3.3.1. Steps 24 and 25 perform the multiset union operation, which may contain repetitive elements like, $\{a, b\} \uplus \{b, c\} = \{a, b, b, c\}$. Steps 6 to 11 build the different metric spaces by constructing different layers. Backtracking from the metric spaces to the original feature space is done with the help of set of indices attached to each layer. It helps to find the proximity vectors, which form the dominant cluster. For each proximity vector, the degree of belongingness to dominant cluster is found using the same distance measure which has been used to construct different metric spaces.

Algorithm 3.3.1 PxC: Proximity Clustering

input: database X, k, number of layers $L \geq 2$, index from the last $1 \leq \mathfrak{i} \leq k$
output: Proximity pairs $\mathbb{P}$

construct ΔX from X
make $d^{(k)}(X)$ by choosing elements from ΔX
$\mathbb{L}_1 = construct(d^{(k)}(X), \mathbb{I})$
set of proximity vectors $P_x = \emptyset$ with degree $P_d = \emptyset$
$lastIndx = N - \mathfrak{i}$
$\backslash *$ *forward traversal by layer formation* $* \backslash$
for layer l from 1 to L **do**
  $\mathbb{L}_l = sortpairset(\mathbb{L}_l)$
  $\left[d_l^{(k)}, \mathbb{I}_l\right] = distruct(\mathbb{L}_l)$
  $d_l'^{(k)} = |\Delta^{(+)}(d_l^{(k)})|$
  $\mathbb{L}_{l+1} = construct(d_l'^{(k)}, \mathbb{I})$
end for
$\backslash *$ *backtrack through the layers* $* \backslash$
for layer l from L to 1 **do**
  $prevIndx = \delta(\mathbb{I}_l, lastIndx)$
  $lastIndx = prevIndx$
end for
$\backslash *$ *collect proximity vectors with their degree* $* \backslash$
for index i from 1 to N **do**
  $properIndx = \delta(\mathbb{I}_1, i)$
  if $properIndx = lastIndx$ **then**
    break
  else
    $P_x = \delta(X, properIndx) \uplus P_x$
    $P_d = (\frac{1-d^{(k)}(P_x)}{1+2d^{(k)}(P_x)})^2 \uplus P_d$
  end if
end for
$\mathbb{P} = construct(P_x, P_d)$

3.4 Experimental Results

The quality of clustering methods is evaluated using *purity* and *entropy*, which are widely used to evaluate the performance of unsupervised learning algorithms [62]. The *purity* measure evaluates the coherence of a cluster. Consider the database having $|DB|$ points of M different categories. Let $\mathbb{C}$ be the cluster of size $|\mathbb{C}|$ and having n^{c_i} points from the category c_i then $|\mathbb{C}| = \sum_{i=1}^{M} n^{c_i}$ and *purity* of $\mathbb{C}$ is given as,

$$P(\mathbb{C}) = \frac{1}{|\mathbb{C}|} \max_{\forall c_i}(n^{c_i}). \tag{3.9}$$

For an ideal cluster having points from a single category, its *purity* is 1. The higher the *purity*, the better the quality of the cluster is. Whereas the *entropy* measure evaluates the distribution of categories within a cluster [63] as,

$$E(\mathbb{C}) = -\frac{1}{log(M)} \sum_{i=1}^{M} \frac{n^{c_i}}{|\mathbb{C}|} log\Big(\frac{n^{c_i}}{|\mathbb{C}|}\Big). \tag{3.10}$$

The averaged entropy is the weighted sum of the individual entropy from all K clusters as,

$$\mathbf{E} = \sum_{i=1}^{K} \frac{|\mathbb{C}_i|}{|DB|} E(\mathbb{C}_i). \tag{3.11}$$

For more detail analysis, the dominant class clustering problem is cast as the classification problem. The data points, which are clustered in the dominant class according to the ground truth, are considered as true positives. This framework allows us to find true positive rate (TPR) and false positive rate (FPR).

For clustering performance comparison, various artificial data points are constructed as shown in 1^{st} column of the Fig. 3.1. 1^{st} row shows the concentric

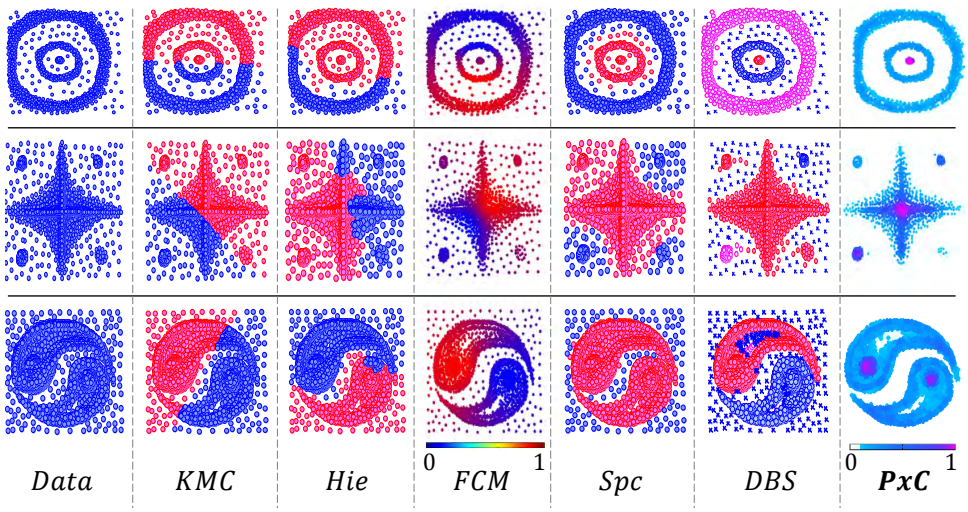


Figure 3.1: Different artificially generated data points and clustering performance comparison of various algorithms.

circles (*concir*) data points, 2^{nd} row is of *star* and 3^{rd} row shows *spiral* pattern. The density of points has also been varied spatially within each data points for evaluating the robustness of the clustering algorithms. In *concir* dataset, the innermost circle has the highest density with almost 60% of the data samples are lying within it. Similarly *star* and *spiral* dataset has the highest density at the central region and the data samples become sparse gradually. Ground truth for these data is that the closely and densely spaced points forming the colony of their own constitute the dominant class and rest of the points are either from rare classes or outliers. So here we can consider there are two clusters viz. the dominant class and class having rest of the other points. Thus to perform clustering via KMC, Hie, FCM and Spc the parameter, #clusters, is set to 2.

For this artificial dataset, Fig. 3.1 shows the clustering results and qualitative performance comparison with other clustering methods. The colorbar associated with FCM and Prx shows the degree of belongingness for each data point to one particular cluster. We can see that Spc, DBS and Prx can solve the dominant

Table 3.1: The performance comparison of the proposed method (Prx clustering) with existing methods for anomaly detection in UMN dataset

Method	Accuracy
Optical Flow [41]	84 %
Social Force [41]	96 %
Chaotic Invariants [42]	99 %
Neural network learning	93 %
Sparse reconstruction cost [40]	97.5 %
Local statistical aggregates [43]	98.5 %
Proximity Clustering (Prx: proposed)	98.1 %

class clustering problem where the performance of DBS and Prx is much better. In the results of Prx clustering, we can see that the degree of belongingness to dominant class is close to 1 for densely spaced samples and 0 for the outliers. Table 3.2 gives the much detail quantitative performance comparison for the same dataset. Quantitatively both DBS and Prx methods are better performing than rest of the methods, while Prx gives the minimum average entropy for all datasets.

For abnormal event detection, the Prx clustering is performed in the feature space constructed from the smoothed features f^{τ} as given in (3.6). The events whose features do not belong or having less degree of belongingness to the dominant class are classified as unusual events. The performance is evaluated on UMN dataset [5], which contains initially normal events and then followed by abnormal ones. In the normal event scenario, people are walking and interacting casually and abnormal activity includes the events like attack of panic, where all people suddenly run apart. Table 3.1, shows the performance comparison of proposed method with the existing algorithms. In addition, the proposed method has an advantage of incremental learning. New unseen event get first classified as unusual event, but if its occurrence is repeated then it will be automatically

Table 3.2: Clustering performance evaluation on artificial dataset and comparison with various methods

Database	Performance Measure	KMC	Hie	FCM	Spc	DBS	Prx
concir	$P(\mathbb{C}_1)$	0.952	0.946	0.944	0.949	0.993	**1.000**
	$P(\mathbb{C}_2)$	0.055	0.046	0.048	0.052	**1.000**	**1.000**
	E	0.293	0.293	0.292	0.293	0.058	**0.000**
	TPR	0.536	0.689	0.430	0.712	**1.000**	**1.000**
	FPR	0.500	0.724	0.474	0.711	0.132	**0.000**
star	$P(\mathbb{C}_1)$	0.931	0.890	0.941	0.956	0.983	**0.996**
	$P(\mathbb{C}_2)$	0.077	0.059	0.086	0.220	**1.000**	**1.000**
	E	0.374	0.369	0.372	0.340	0.117	**0.037**
	TPR	0.613	0.253	0.507	0.864	**1.000**	**1.000**
	FPR	0.586	0.400	0.407	0.507	0.221	**0.057**
spiral	$P(\mathbb{C}_1)$	0.963	0.956	0.959	0.999	**1.000**	**1.000**
	$P(\mathbb{C}_2)$	0.034	0.034	0.036	**1.000**	0.649	0.958
	E	0.234	0.233	0.234	0.015	0.054	**0.009**
	TPR	0.497	0.414	0.427	**1.000**	0.978	0.998
	FPR	0.478	0.478	0.461	0.026	**0.000**	**0.000**

classified as normal event in subsequent video clips. That is the method self learns to the new normal events in an unsupervised manner.

3.4.1 Improving image classification

Bag-of-Features (BoF) captures invariance aspects of the local image features by depicting their orderless collection. It has shown excellent performance for various visual classification tasks. Dictionary learning is nothing but obtaining the compact representation of BoF through clustering of features from all dataset images. These clusters are 'visual words', which form the visual codebook. Each image is described as the normalized histogram of codebook entries. This BoF representation is treated as the new image feature vector (IFV). This procedure does not consider class labels of the images; which are used, only after getting IFV, for image classification using supervised learning methods.

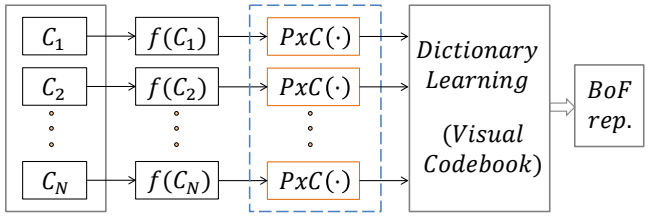


Figure 3.2: System for obtaining bag of features representation with PxC

PxC helps to embed the class label information for obtaining IFV as shown in Fig. 3.2. Let the database contains images from N different classes $\{C_1, C_2, \cdots, C_N\}$. The function $f(C_i)$, extracts various features from all the images of class C_i. And these features are semantically similar as they are from the same image class. So as described in sec. 6.1 and 3.3.1, PxC finds a porous cluster in the form of proximity vectors by removing outliers and infrequent patterns. Then all proximity vectors are sent for the dictionary learning, which is followed by the procedure of obtaining IFV.

Image classification experiments were performed on Caltech categories [64] dataset containing total 3188 images from 5 different classes. Filter bank is constructed as in [65], for extracting various image features. Different multiclass classification tasks were performed for properly assessing the benefits of the PxC. Fig. 3.3 shows the average classification accuracy obtained after 10 fold cross validation. It shows that the use of PxC for obtaining IFV, improves the classification accuracy of each classification task.

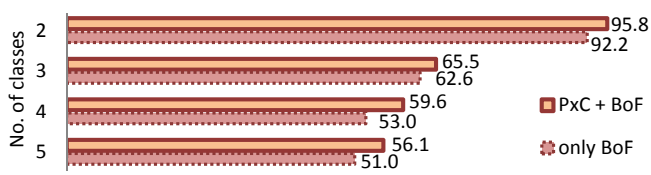


Figure 3.3: Improvement in image classification accuracy (%) using PxC

Figure 3.4: Unsupervised image segmentation using PxC and qualitative comparison with k-means clustering segmentation. (*best viewed after zooming in*)

3.4.2 Improving unsupervised image segmentation

Training images are not always available to perform robust segmentation. So synthetic aperture radar (SAR) images, medical images, texture dominant, satellite and telescopic images are need to be segmented in an unsupervised manner. Segmentation performance can be improved by first performing PxC over CIE Lab color space of an image, which helps to retain the coherent colors. Then KMC is performed over proximity vectors to get k clusters, which form the k segmented parts of an image. After this the feature points from each cluster are represented with their spacial co-ordinates corresponding to original image and again PxC is performed on them. It smooths out the segmented parts by making them compact and thereby improves the segmentation accuracy.

Fig. 3.4 (a) shows the original images; (b) and (d) are their segmented parts (background and foreground respectively) by using KMC alone; whereas (c) and (e) segments are obtained by alternatively employing PxC on two different feature spaces along with KMC as explained previously. Segments (b), (d) are noisy, firstly because (b) contains extraneous foreground information which should be present in (d) and, secondly they are speckled. Whereas (c), (e) are less noisy, containing much vivid colors and continuous parts.

3.5 Conclusion

In this work we have solved the abnormal event detection in surveillance video problem by proposing the Proximity (Prx) clustering algorithm. It performs the dominant class clustering and rejects the rest of the points which are either from sparse classes or outliers. We also have proposed the motion features namely circulation, motion homogeneity, motion orientation and stationarity, which help to extract the relevant information from the scene necessary for abnormal event detection.

The quantitative and qualitative clustering performance comparison on artificial dataset shows that the Prx clustering outperforms the other existing algorithms for single dominant class clustering. It also has an advantage that for each data sample, it gives the degree of belongingness to the dominant cluster. This helps to produce the flexible porous cluster structure of the dominant class. Prx clustering in the motion feature space gives the easy way to detect the abnormal events with comparable detection accuracy with the existing methods. It also has capability of incremental learning that it learns about the new normal events in an unsupervised manner.

The Prx clustering is promising for outlier rejection when the sample points belong to only one class. More motion features can easily be added in the proposed framework to improve the anomaly detection performance. But there is trade off between the computation time and the performance, since analysis of video events in higher dimensional feature space is much more complicated and extracting more number of features will also take longer time. Our future work will try to address these issues and opportunities.

Chapter 4

Graph Pyramid: A hierarchical graph analysis framework

4.1 Introduction and Prior Work

Human action and gesture recognition are crucial facets of an automated video surveillance system, because they assist in understanding the semantic concepts and the subtle patterns of the human behavior. A rich palette of numerous ideas have been put forward over the last decade on human activity recognition (HAR) by employing diverse visual information. Graphical modeling is a promising approach for uncovering the hidden activity patterns from the video, as it gives the capability to efficiently assess all possible interactions among the graph nodes.

For HAR involving interactions between people, [66] has used the graph-based spectral matching of the local features and [67] has modeled the entire scene as an error-free graph with each node corresponds to a patch of the scene and an edge represents the activity correlation between the patches. They inherently lack in semantic activity information as they use primitive and mid-level features. Also,

building a graph for each temporally evolving motion sequence and performing graph matching are computationally very expensive operations. Individual HAR has been performed by encoding the actions in a weighted directed graph using salient body postures as a graph nodes [68]. Whereas [69] has constructed a star skeleton graph using body contours and [70] has performed graph modeling of temporal causal relations of the body joint movements which requires motion capture data. [71] and [72] have encoded the spatio-temporal interest points and SIFT local features in-to the graph respectively. These features need to be extracted only within tight neighborhood of the person, so they require precise tracking information. Needs of salient detection and tracking, usually make local feature based approaches to meet limitations in real-life applications, where obstacles like occlusions, scarce computational resources, viewpoint and illumination changes are ubiquitous. Whereas holistic approaches produce more robust features at lower computational cost. So our approach follows this vein for video representation.

Unlike other methods, we encode entire class of the activity in the graph topology, where each node represents the sample motion sequence from the same class. In this modeling, the test sample is just another graph node, so graph matching is not necessary for making inference. Methods based on similarity clustering [73], k-nearest neighbors and phylogenetic clustering [74] just look for closely interacting nodes nearby the query. But they fail to account for interactions among the closely interacting neighborhood of the query, which leaves the room for performance improvement. Also, individual variability of the actions due to changes in the background, actors, illumination and the viewpoints introduce subtleties in each action class.

So we propose a graphical model for HAR. Unlike other methods, we construct the graph of the entire action class by representing each motion sequence

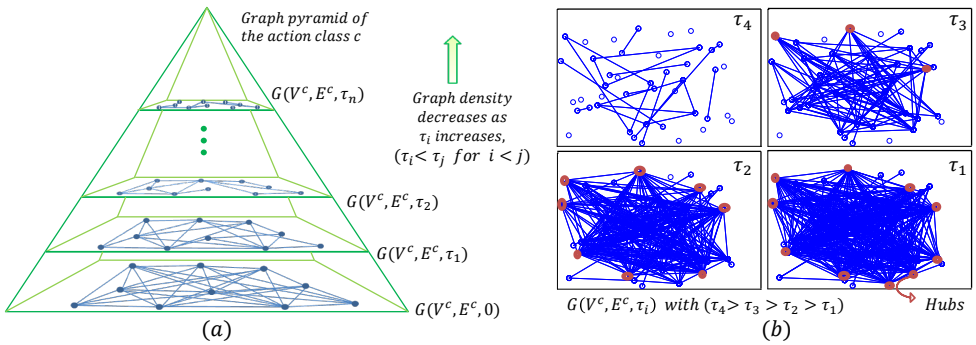


Figure 4.1: (a) Graph Pyramid (GP): a hierarchical graph analysis, (b) Building of Action-Action Similarity (AAS) network with varying threshold shows the formation of hubs. Here class c is formed by the samples from action *boxing* in KTH [6] dataset.

as a node. Training embeds the class specific information in the graph edges via our modified quadratic-Chi distance. Hierarchical graphical analysis makes it possible to uncover the hidden subtleties of the action family by considering interactions among the neighborhood nodes to the query. Some of the salient features of the proposed method include, incremental training capability and computational efficient inference about the query.

4.2 Method and Graph Modeling

Holistic features like motion energy images [75], template based representation [76] and Zernike moments [77] can succinctly describe the entire motion sequence. However bag-of-words based modeling can produce the global action representation, even by using local (silhouette, gesture, optical flow) [77] and mid-level (spatio-temporal cuboids, object trajectories) features. These methods produce a unique histogram per action sequence (denoted as s_i) by keeping account of occurrences of all dictionary words or templates.

4.2.1 Graph Construction

The proposed algorithm uses graphical modeling of the action sequences from each action class. Let the set of class labels for the action database having total M classes be $\mathbb{C}_M = \{c_1, c_2, c_3, \cdots, c_M\}$. For class $c \in \mathbb{C}_M$, let $s^c_{t_i}$ be the i^{th} training action sequence. Similarly $s^{c_q}_q$ is the new query sequence whose class label c_q is what we have to infer. This part is addressed later in the algorithm section 4.2.4.

Let class c has N training sequences, then the set of vertices in similarity space is defined as, $V^c = \{s^c_{t_1}, s^c_{t_2}, \cdots, s^c_{t_N}\}$. Strength of an edge between the vertices $s^c_{t_i}$ and $s^c_{t_j}$ is given by the quadratic Chi distance [78] between them, like $e^c_{i,j} = \mathcal{QC}^A_m(s^c_{t_i}, s^c_{t_j})$; where $0 < m < 1$ is normalization factor and A is the feature dimension similarity matrix. For a particular class c, some group of feature dimensions produce high or low response together and consistently over all samples. So to make use of this inherent dynamics among feature dimension, we propose a new similarity matrix tailored towards each action class. Let mean feature of c be $\bar{s}^c_t = \frac{1}{N}\sum_{i=1}^{N} s^c_{t_i}$, then for the same class,

$$A^c[i,j] = 1 - \frac{|\,\bar{s}^c_t[i] - \bar{s}^c_t[j]\,|}{\max(\bar{s}^c_t[i], \bar{s}^c_t[j])}. \tag{4.1}$$

This similarity matrix is pre-computed via training, so $e^c_{i,j}$ computation is linear in the non-zero entries of A^c. These edges form the set $E^c = \{e^c_{1,1}, e^c_{1,2}, \cdots, e^c_{N,N}\}$. Now graph of the action class $c \in \mathbb{C}_M$ is given by $G(V^c, E^c)$. This is a weighted and an undirected graph. An edge weight is nothing but the degree of action similarity. For constructing $G(V^c, E^c)$, we just need to consider action interactions within only class c. Number of samples in a class is a way smaller than that of entire database. So graphs of all classes can be easily and independently constructed, which are called as Action-Action-Similarity (AAS) networks $\big($see Fig.4.1$(b)\big)$.

4.2.2 Graph Analysis

In action similarity graphs, modularity, local clustering and scale-free topology coexist, due to the presence of multiple viewpoints and repetition of similar motions. To explain this phenomenon we need the hierarchy, so graphs are analyzed in hierarchical manner. At each hierarchical level, the edges with weights lower than certain threshold are pruned. After this the survived edges are considered to be weightless. So the graph topology changes along hierarchical levels, where the graph becomes unweighted and remains undirected. It helps to extract different graph features for even weakly similar hits (action matches) and thus captures the complex relationship within the action class.

For any set $\mathbb{S}$, let $c \in \mathbb{S}$, $\hat{\imath}$ be an indicator variable and the null set, $\emptyset = \{\}$ then 'set element' is formed as, $\delta(c, \hat{\imath}) = \{c \text{ if}(\hat{\imath} = 1) \,|\, \text{else } \emptyset\}$. Cardinality of a set ($|\mathbb{S}|$), is the number of elements it has. Let $H(x) = \{1 \text{ if}(x > 0) \,|\, \text{else } 0\}$ be the Heaviside-step function, then for a graph of c at certain hierarchy (i.e. at threshold τ), the edge set is given as,

$$\lceil E^c \rceil_\tau = \bigcup\nolimits_{e^c_{i,j} \in E^c} \delta(e^c_{i,j}, H(e^c_{i,j} - \tau)). \tag{4.2}$$

For notation simplicity lets represent the corresponding graph as $G(V^c, E^c, \tau)$ instead of $G(V^c, \lceil E^c \rceil_\tau)$, and note that $G(V^c, E^c) = G(V^c, E^c, 0)$ $\big($see Fig.4.1$(a)\big)$. Let the multiset union operation for any sets be $\mathbb{S}_1 \uplus \mathbb{S}_2 = \{\{\mathbb{S}_1 \cup \mathbb{S}_2\}, \{\mathbb{S}_1 \cap \mathbb{S}_2\}\}$ and note that $|\mathbb{S}_1 \uplus \mathbb{S}_2| = |\mathbb{S}_1| + |\mathbb{S}_2|$, thus united set contains repetitive elements when $\{\mathbb{S}_1 \cap \mathbb{S}_2\} \neq \emptyset$. It obeys associative and commutative laws like numerical addition. After adding the query s^{cq} to the original graph $G(V^c, E^c)$, we will get the new graph $G(V^c_q, E^c_q)$, where $V^c_q = V^c \uplus s^{cq}_q$, and the edges among vertices in the set V^c_q are given by E^c_q.

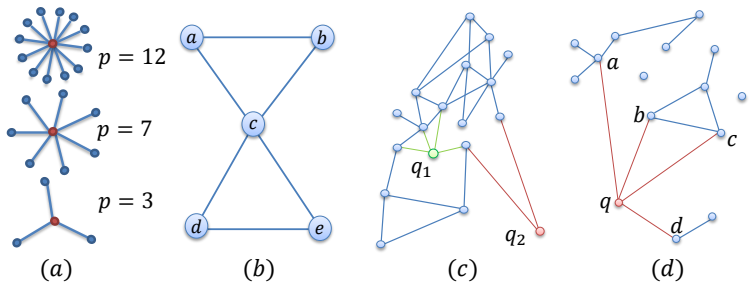


Figure 4.2: Explanation for GTFs. (a) SMs with different powers, (b) Example graph to explain AC and RC, (c) Different queries (q_1, q_2) affect the graph spread (GE) differently, (d) TR and SM capture different network properties.

4.2.3 Graph Topological Features (GTF)

Most of the real world and biological (scale-free) networks communicate via few highly connected nodes known as Hubs. These hubs determine network's properties [79]. In real world networks like airline route map, the important cities form hubs. In case of human activities, some people have peculiar way of performing motions (e.g. gait of a person). But to accomplish particular action (*walking*) they always have to go through specific limb movements (moving a pair of legs in turn). So same action class sequences show high degree of connectedness in the similarity space, which shows the possibility of hub formation in $G(V^c,E^c)$.

Fig.4.1(b) shows the building of AAS network with varying threshold for *boxing* action from KTH dataset. We can see that as the threshold is lowered, more hubs are starting to build up. We are not interested in the detail assessment of whether the network is scale-free (a power-law degree distribution [79]) or not. But above analysis helps to guide us for finding proper features which take graph topology (i.e. complex relationships among action sequences) in to account. Also, different action families have different characteristics. Thus use of single graph feature may not be effective. Features are selected such that they could extract uncorrelated as well as vital information from the network.

• **Average Clustering coefficient** (AC): For a node n, the clustering coefficient C_n, measures the extent to which neighbors of n are also neighbors of each other [79]. Thus it is nothing but the density of sub-graph induced by the neighborhood of n. For the graph $G(V,E)$ with $n \in V$, let $\mathbb{N}_n$ be the number of neighbors of n and $\mathbb{E}_n$ is the number of connected edges between them, then the AC is given by,

$$AC\big(G(V,E)\big) = \frac{1}{|V|} \sum_{n \in V} \left(\frac{2\mathbb{E}_n}{\mathbb{N}_n(\mathbb{N}_n - 1)} \right). \tag{4.3}$$

A clique is a maximal complete sub-graph where all the vertices are connected. C_n quantifies how close the node's neighbors are to form a clique among themselves. It represents a network's potential modularity and C_n of the most real networks is much larger than that of a random network. AC distribution is found to be effective for an identification of modularity in the metabolic networks [80]. Consider the example shown in Fig.4.2(b). Node c has 4 neighbors ($\mathbb{N}_c$), having 2 connected edges ($\mathbb{E}_c$) among them (a to b and d to e), which forms $C_c = \frac{1}{3}$ and then $AC = \frac{13}{15}$. When s_q^{cq} is attached to $G(V^c,E^c)$, it may change its AC. For a given s_q^{cq} the change in AC at threshold $\tau \in \tau_{AC}$ is given as,

$$\Delta AC(c,\tau) = AC\big(G(V_q^c,E_q^c,\tau)\big) - AC\big(G(V^c,E^c,\tau)\big). \tag{4.4}$$

• **Rich Club coefficient** (RC): The 'rich-club' phenomenon refers to the tendency of nodes with high centrality to form tightly interconnected communities. Degree (d) of a node is the number of directly connected neighbors. High degree nodes (*rich nodes*) are much more likely to form tight and well interconnected sub-graphs than low degree nodes [81]. Thus hubs are generated through 'rich-gets-richer' mechanism. A quantitative definition of the rich-club phenomenon is given by the rich-club coefficient (ϕ). Let $\mathbb{N}_{d>r}$ be the number of vertices having

degree greater than r and $\mathbb{E}_{d>r}$ be the number of edges among them then,

$$RC\big(G(V,E),r\big) = \phi(r) = \frac{2\mathbb{E}_{d>r}}{\mathbb{N}_{d>r}(\mathbb{N}_{d>r}-1)}. \tag{4.5}$$

For the example in Fig.4.2(b), $\phi(1) = \frac{3}{5}$. In a complex network, ϕ is a novel probe for finding topological correlations and it yields vital information about network's underlying architecture [81]. Similarly as explained earlier, the change in RC at threshold $\tau \in \tau_{RC}$ is given as,

$$\Delta RC(c,\tau,r) = RC\big(_{G(V_q^c,E_q^c,\tau)},r\big) - RC\big(_{G(V^c,E^c,\tau)},r\big). \tag{4.6}$$

• **Star Motifs** (SM): Previous analysis showed the existence of hubs in AAS network. Features like mean path length and degree distribution [79], nicely quantifies hub like properties of the network. But they are computationally expensive for large action database, so simple and elegant features are needed. Hubs indicate existence of star shaped patterns (*star-motifs*). Define, power (p) of SM as the degree of the centered node. Fig.4.2(a) shows the SM with various powers. Each node in the training graph is already assigned with node degree, so that SM can be easily computed. This simple feature answers the questions like, 'does $s_q^{c_q}$ give rise to new hubs' and 'what is the increased strength of hubs'. Consider $SM(\cdot,p)$, which finds the number of SM of power p then change in SM at threshold $\tau \in \tau_{SM}$ is given as,

$$\Delta SM(c,\tau,p) = SM\big(_{G(V_q^c,E_q^c,\tau)},p\big) - SM\big(_{G(V^c,E^c,\tau)},p\big). \tag{4.7}$$

• **Triangles** (TR): In a graph, the smallest clique with three nodes is a triangle. The number of triangles, gives an important information about structure of the network. Let the graph $_{G(V^c,E^c)}$ be represented by $N\times N$ adjacency matrix $[E^c]$, whose $(i,j)^{th}$ entry is given as, $[E^c](i,j) = H(e_{i,j}^c)$, then the number of triangles

in the graph are given as,

$$TR\big(G(V^c, E^c)\big) = \frac{1}{6}\sum_{i=1}^{N} [E^c]^3\,(i,i). \tag{4.8}$$

Either the graph is dense or sparse, TR can be computed readily within the same time. TR inherently captures different network properties than SM. Formation of new triangles in a graph indicates the fact that 'query interacts simultaneously to the already interacting nodes'. Fig.4.2(d) illustrates this difference more elaborately. Query node q interacts with nodes a, b, c and d. After interaction, 1 triangle is formed since only nodes b and c are previously interacting and at node a the SM power has increased from 3 to 4. Due to query interaction, the newly formed number of triangles $\Delta TR(c,t)$, are found similarly like (4.4).

• **Graph Energy** (GE): The original graph structure changes when query interacts with it. Fig.4.2(c) shows two different type of query interaction with the same graph, where edge length is proportional to its weight. When only q_1 interacts with a graph then its spread remains almost unaffected but in case of q_2 interaction, graph's spread alters drastically. GE is defined as the sum of the absolute eigenvalues of the adjacency matrix [82], and given as follows,

$$GE\big(G(V^c, E^c)\big) = \sum_{i=1}^{N} \lambda_i\Big([E^c]\Big). \tag{4.9}$$

Since GE depends only on adjacency matrix, the graph density doesn't affect its computation time. The effect of query interaction on the original graph is captured by the change in GE ($\Delta GE(c,t)$), given similarly like (4.4). We are dealing with graphs whose edge strength is the similarity between connecting nodes, which is inverse of the usual edge length definition. So we need to look for maximum $\Delta GE(c,t)$.

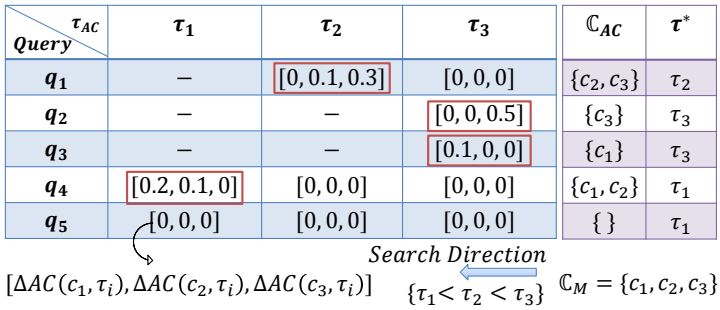


Figure 4.3: Illustration of the algorithm 4.2.1 for 5 different queries. For simplicity assume $T_{AC} = 0$, $\tau_{AC} = \{\tau_1, \tau_2, \tau_3\}$ and only 3 action classes i.e. $|\mathbb{C}_M| = 3$. Each q_i is analyzed first at the highest level (τ_3) where we look for all classes $c \in \mathbb{C}_M$, having $\Delta AC(c, \tau_3) > T_{AC}$ and collect them in $\mathbb{C}_{AC} \subseteq \mathbb{C}_M$. For q_1 we can not find any such classes c at τ_3, so we descend the GP to τ_2 level and discover $\mathbb{C}_{AC} = \{c_2, c_3\}$ with $\tau^* = \tau_2$. This secondary threshold (T_{AC}) is necessary, otherwise there will be many spurious classes (false positives (FPs)) having nonzero $\Delta AC(\cdot)$.

4.2.4 Algorithm

Graphs are analyzed hierarchically (sec.4.2.2) and threshold plays an important role in making hierarchical graph structure. If the query can interact at the higher layer of GP (see Fig.4.1(a)), then it means its a strong interaction. Because in GP as the level rises, threshold also increases; and at every level, graph edges can only be formed if their strength is greater than the given threshold. Let $\mathbb{T} = \big\{\{\tau_{AC}\}, \{\tau_{RC}\}, \{\tau_{SM}\}, \{\tau_{TR}\}, \{\tau_{GE}\}\big\}$, be a set of sets, defining few threshold levels corresponding to each GTF. Fig.4.3 explains the GP search subroutine (algorithm 4.2.1) for the feature AC. Where, τ^* is the maximum threshold (also defines maximum GP level) at which query starts interacting with some of the AAS networks. Let's represent this subroutine by abusing notation for simplicity as,

$$\mathbb{C}_{AC} \longleftarrow \underset{c \in \mathbb{C}_M}{\operatorname{argmax}}\, H\big(\Delta AC(c, \tau^*) - T_{AC}\big). \tag{4.10}$$

Maximum value of $H(\cdot)$ is 1, so all the arguments (classes c) are assigned to $\mathbb{C}_{AC}$ whenever it produces output 1. For reducing FPs, the subroutine for SM and TR

Algorithm 4.2.1 Graph pyramid (GP) search subroutine

input: $s_q^{c_q}$; secondary threshold T_{AC}; primary threshold $\tau_{AC} = \{\tau_1, \tau_2, \cdots, \tau_n\}$, $\tau_i > \tau_j$ for $i > j$

output: $\mathbb{C}_{AC}$, τ^*

```
1: C_AC = ∅
2: for τ ∈ τ_AC from τ_n to τ_1 do
3:    if C_AC = ∅ then
4:       for all classes c ∈ C_M do
5:          I_AC = H(ΔAC(c, τ) − T_AC)
6:          C_AC = C_AC ⊎ δ(c, I_AC)
7:          τ* = τ
8:       end for
9:    end if
10: end for
```

slightly changes (see algorithm 4.2.2). Here we look for k maximally influenced classes from $\mathbb{C}_{RC}$ by the query. Thus classes only from $\mathbb{C}_{RC}$ are assessed (voted) again by the features SM and TR. Subroutine for GE finds the class from $\mathbb{C}_{SM} \cup \mathbb{C}_{TR}$ for which $\Delta GE(\cdot)$ is maximum. Thus this produces hierarchical voting scheme (algorithm 4.2.2) which helps to improve HAR rate and to reduce computational load. Each GTF has an ability to extract different information from different levels of the action family GP. However, applying each GTF to entire GP, is computationally inefficient when dealing with large number of action families. In addition, it may add up the FPs, when decision is being made at much lower GP level than the level defined by τ^*. To avoid these issues, the GP based hierarchical voting is necessary. And the rational behind placing different GTFs at different GP levels, is explained in the experimental sec.4.3.

4.3 Experimental Results

Datasets and evaluation details. We evaluated the proposed approach on publicly available three action and one gesture benchmark datasets, viz. UCF-

Algorithm 4.2.2 Classification by hierarchical voting

training: all $G(V^c, E^c)$ are constructed $\forall c \in \mathbb{C}_M$
input: $s_q^{c_q}$, r, p, k, T_{AC}, T_{RC} and threshold set $\mathbb{T}$
output: c_q

$\mathbb{C}_{AC} \longleftarrow \text{argmax}_{c \in \mathbb{C}_M} H\big(\Delta AC(c, \tau^*) - T_{AC}\big)$
$\mathbb{C}_{RC} \longleftarrow \text{argmax}_{c \in \mathbb{C}_{AC}} H\big(\Delta RC(c, \tau^*, r) - T_{RC}\big)$
$\mathbb{C}_{SM} \longleftarrow \arg\ \text{k-max}_{c \in \mathbb{C}_{RC}} \Delta SM(c, \tau^*, p)$
$\mathbb{C}_{TR} \longleftarrow \arg\ \text{k-max}_{c \in \mathbb{C}_{RC}} \Delta TR(c, \tau^*)$
$\mathbb{C}_{GE} \longleftarrow \text{argmax}_{c \in \mathbb{C}_{SM} \cup \mathbb{C}_{TR}} \Delta GE(c, \tau^*)$
$\psi_q = mode\big(\mathbb{C}_{AC} \uplus \mathbb{C}_{RC} \uplus \mathbb{C}_{SM} \uplus \mathbb{C}_{TR} \uplus \mathbb{C}_{GE}\big)$
if $|\psi_q| \geq 2$ **then**
    $\mathbb{C}_{SM} \longleftarrow \text{argmax}_{c \in \mathbb{C}_{RC}} \Delta SM(c, \tau^*)$
    $\mathbb{C}_{TR} \longleftarrow \text{argmax}_{c \in \mathbb{C}_{RC}} \Delta TR(c, \tau^*)$
    $\psi_q^{'} = mode\big(\mathbb{C}_{SM} \uplus \mathbb{C}_{TR} \uplus \mathbb{C}_{GE}\big)$
    if $|\psi_q^{'}| \geq 2$ **then**
        $c_q = \mathbb{C}_{GE}$
    else
        $c_q = \psi_q^{'}$
    end if
else
    $c_q = \psi_q$
end if

sports [7], KTH [6], Weizmann [4] action datasets and HWU [3] gesture dataset. They consist of different types of actions, some of which are delineated in Fig.4.4 by labels of the corresponding confusion matrix. After cross validation, we set, $p^* = 2$ for all GP levels, while r^* is 3 for lower and 5 for higher GP levels. We used template based global representation [76] for each action sequence and followed the same experimental setup as described by the dataset publishers.

Building the pyramid. First, each GTF was tested independently for various thresholds, where the class producing maximum change in that GTF for a given $s_q^{c_q}$, was selected as the output. These output labels were produced with either correct decision (cd), wrong decision (wd) or no decision (nd when $|c_q| \neq 1$). Then, $precision = \frac{cd}{cd+wd}$. Fig.4.4($a$) shows the plots of precision vs

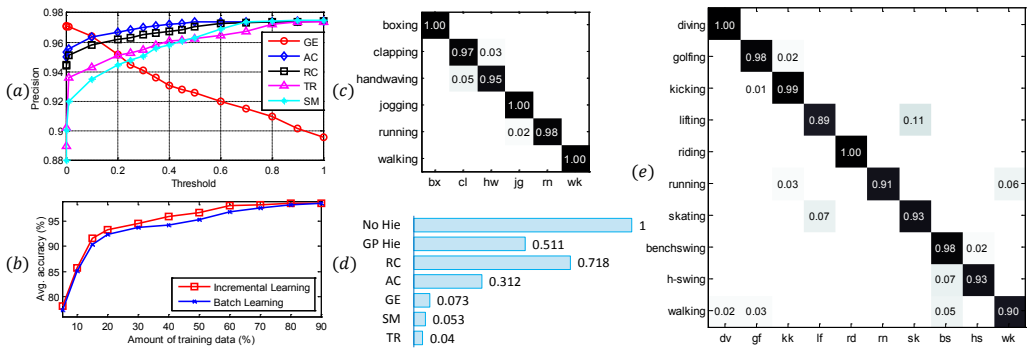


Figure 4.4: (a) Precision vs threshold for all GTFs, (b) Incremental learning advantage, (c) Confusion matrix (KTH [6]) , (d) Normalized computational time for GTFs, their collective decision with and without GP hierarchy (GP Hie, No Hie), (e) Confusion matrix (UCF-sports [7] dataset).

threshold for all GTFs. High precision indicates low wrong decisions. AC and RC produce high precision as threshold rises, so these GTFs are appointed to work at higher levels of GP. So at high threshold, it is more likely that $\mathbb{C}_{AC}$ and $\mathbb{C}_{RC}$ contain true output class. Thus it is sufficient to apply other GTFs, to the GPs generated from either $\mathbb{C}_{AC}$ or $\mathbb{C}_{RC}$. On the other hand, GE produces high precision for low threshold. One of the possible reasons behind this is that, the higher the threshold, the sparser will be the graph. So eigen-decomposition of the adjacency matrix to calculate GE will not give any information as ΔGE is close to zero for all classes. While at low threshold original graph becomes dense and query also interacts with almost all nodes in the true output class graph. Immensity in the interaction at lower threshold, helps GE to detect true class easily and correctly. This forces GE to work at lower levels of GP, with the smallest search space as $\mathbb{C}_{SM} \cup \mathbb{C}_{TR}$. With similar arguments, SM and TR are placed at intermediate GP levels, allowing them to look for c_q in $\mathbb{C}_{RC}$. So in the algorithm 4.2.2, hierarchy as well as input search space for GTFs are organized carefully. This helps to reduce the search space dramatically for other GTFs and thus speeds up the algorithm (see Fig.4.4(d)). Sometimes, for $s_q^{c_q}$ having subtle

interaction with many classes, it is difficult for all GTFs to come up with an unique agreement about c_q. When it happens, the threshold would have already hit the bottom of its range. Thus, with earlier reasoning, the solution would be to rely only on GE to find c_q, and 12^{th} step in the algorithm 4.2.2 does the same.

Performance analysis. KTH and UCF-sports are more complex and challenging datasets, including a wide range of variations in background and viewpoints. So their detailed HAR analysis for each activity is given in the form of confusion matrices in Fig.4.4(c) and (e) respectively. On UCF-sports, the proposed method produces the highest 95.1% accuracy, where it slightly interchangeably confuses between *lifting* and *skating*. It also recognizes gesture on HWU dataset with 98.2% accuracy. Table4.1 compares performance of the state-of-the-art methods on all HAR datasets.

Incremental learning. We add $s_q^{c_q}$ back to c_q after its classification in the KTH dataset. This is an instance based learning and it only requires slight modification of earlier c_q graph, like replacing $G(V^{c_q}, E^{c_q})$ with $G(V_q^{c_q}, E_q^{c_q})$. We select various (5 to 90) percent data from each $c \in \mathbb{C}_M$ for training purpose. Fig.4.4(b) shows the average classification performance comparison between incremental and batch learning. Incremental learning takes an advantage of the each correctly classified $s_q^{c_q}$ and produces better performance even during the scarcity of the training data.

Computational time. Fig.4.4(d) shows the normalized time taken by different classifiers on KTH dataset. In majority voting ('No Hie'), first all GTFs classify each sequence from the large pool of testing sequences, and then the voting begins. This slows down the scheme and also introduces FPs. Whereas, in the proposed GP hierarchical scheme ('GP Hie'), testing pool is gradually shrunken down. So the subsequent GTFs have to investigate only small set of sequences

Table 4.1: Different HAR methods and their average accuracy (%), for HWU gesture, Weizmann, KTH and UCF-sports datasets

Method	**HWU**	Method	**Weizmann**
SVM [36]	88.9	Campos et al. [38]	96.7
Barattini et al. [3]	78.0	Cai et al. [83]	**98.2**
Bomma et al. [36]	98.0	Luo and Hu [17]	96.7
Ours	**98.2**	Ours	98.0

Method	**KTH**	**UCF-sports**
Wang et al. (2009) [10]	89.00	83.3
Campos et al. (2011) [38]	91.50	80.0
Sadanand et al. (2012) [76]	98.20	95.0
Cai et al. (2013) [83]	94.20	91.6
Ours (Graph Pyramid)	**98.33**	**95.1**

which likely to contain the true action class. This in turn speeds up the proposed method along with maintaining high accuracy. Being twice faster than 'No Hie', with maintaining high performance, the 'GP Hie' method is the preferable solution for time consuming classification tasks.

4.4 Conclusion

Instead of modeling each activity separately like existing methods, we model the entire activity class graphically (AAS network) using holistic representation and modified quadratic-Chi distance. The proposed method exploits the structural information of the AAS network using important topological features via hierarchical network analysis guided by the graph pyramid. Thus necessary information for HAR from weak interactions in AAS network is not suppressed by the other strong ones. Additionally it can show topologically how query interacts with the action family and has the incremental learning capability and high HAR rate.

Chapter 5

Handling Imbalanced Datasets: G-SMOTE Algorithm

5.1 Introduction and Prior Work

In the domain of pattern recognition, the classification tasks are of paramount importance. For classification, the learning algorithms are developed to uncover the hidden subtleties from training datasets. This helps to gather information about the trends of each class. These trends are used in predicting the class labels for new instances. The prediction performance depends on the underlying assumption that training set is evenly distributed. But when this distribution of the input samples is skewed, the class imbalance problem arises!

In real-world scenario, abnormalities (interesting events) happen rarely. So often the datasets are predominantly composed of 'normal' instances with only a small percentage of 'abnormal' ones. A dataset is said to be imbalanced, when one class (the minority class) is heavily under-represented in comparison to other (the majority class). When the classifiers are fed with imbalanced datasets, it

causes them to become biased towards majority class. This is a serious problem because the cost of miss-classifying an abnormal instance as a normal is often much higher than the cost of the reverse error.

Occurrence of high imbalance in real-world domains is a direct result of rarity of interesting events, which results in skewed datasets. Without dataset rebalancing, the learning algorithm will encounter extremely low minority class samples therefore it gets biased towards the majority class in the classification tasks. A number of application domains exists where a massive disproportion in the size of the classes is common. For example, credit card transactions (legitimate transactions are much higher than fraudulent ones), medical diagnoses, large protein family classification [84], abnormal event detection in surveillance videos [18], oil spills in satellite images [85], etc.

The class imbalance problem has been solved both at the algorithmic and data levels. At the algorithmic level, solutions include adjusting the various algorithmic parameters (e.g. decision threshold) to increase the sensitivity towards the minority class for constructing the proper decision boundary for classification; recognition based (1 class) learning rather than discrimination based (2 class) learning [86]; mixture-of-experts approach [87]; modified neural network [88]; improved decision tree [89] and naïve Bayes methods [90]. Though these approaches work well, they are designed and tuned for solving special domain problems and face difficulty in generalization.

At the data level solution to the class-imbalance problem include different data sampling techniques. Random undersampling is a non-heuristic method that tries to balance the class distribution via randomly eliminating the majority class instances [91]. But it may discard potentially useful data that could be important for training the classifier. On the other hand, random oversampling method tries to balance the class distribution through random replication of the minority

class instances [92]. As it makes exact copies of the minority class samples, it can lead to overfitting [93]. Another framework of feature sampling [94], selects the features for positive and negative classes separately and then recombines them to get good classification performance with imbalanced data. But this method highly dependent on the type of data under consideration. SMOTE generates the synthetic minority instances by oversampling the minority class [95], which works very well for many applications. This oversampling is performed by randomly interpolating between the several minority class instances that lie together (nearest neighbors). Random imputation within vicinity of the original sample avoids the overfitting problem but sometimes causes over-generalization. It blindly populates the minority area without regard to the majority class and causes decision boundaries for minority class to spread further into majority class [92], thereby mixing the classes. For highly skewed data distributions where minority class is very sparse, it may form the class mixture with a greater chance.

We improved the SMOTE approach by introducing hybrid sampling, which is partially guided by the uncovered hidden patterns from minority class to prevent over generalization; and partially carried out by randomization to prevent the overfitting. Bootstrapping with simultaneous over and undersampling is used to handle the highly skewed data distributions. We also give the brief discussion about importance of the dataset rebalancing techniques in sec. 5.4.

5.2 Algorithmic Framework

First we consider the Gaussian noise distribution which is suitable for the datasets having less intra-class variability. Then for datasets having nonlinear data patterns, we consider the kernel functions to uncover its low-dimensional subspace structure.

5.2.1 Gaussian model of data generation

Let $X_m = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$ be the $d \times N$ matrix of d dimensional N data points of minority class. Suppose the data are generated as, $\mathbf{x}_i = \mathbf{y}_i + \mathbf{e}$ and $\mathbf{x}_i \in \mathbb{R}^d$, where $\mathbf{y}_i$ is the original data vector which lies on a low-rank subspace and $\mathbf{e}$ is the error term and $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \cdots, x_{i,d}]^T$. A degenerate Gaussian distribution can properly model a low-rank subspace [96]. This low rank data projection can be learned by minimizing the overall reconstruction error [97] as,

$$\min_{U} \|X_m - UU^T X_m\|_F^2,$$
$$\text{subject to} \qquad U^T U = \mathbb{I}_r, \tag{5.1}$$

where $\mathbb{I}_r$ is an identity matrix of size $r \times r$, $\|\cdot\|_F$ is the Frobenious norm of a matrix and r is the rank of U. The solution of (5.1) say $\hat{U}$, can be efficiently given by singular value decomposition of X_m. The columns of U contain the eigenvectors ($\mathbf{u}_i$) of $X_m X_m^T$ with corresponding eigenvalues λ_i. Then the linear principal component is given as,

$$H_l = \{\mathbf{u}_i \,|\, i = \underset{i}{\operatorname{argmax}}\, \lambda_i\}. \tag{5.2}$$

So for a given data or feature sample $\mathbf{x}_i$, its original vector $\mathbf{y}_i$ can be estimated like $\mathbf{y}_i = \hat{U}\hat{U}^T\mathbf{x}_i$. This model helps to uncover the low dimensional subspace structure from high dimensional corrupted observations, when the data are best described by second order correlation or when it is generated by Gaussian distribution. Among all orthogonal linear projections, the principal component projection minimizes the squared reconstruction error [97]. Thus when $\mathbf{e}$ follows the Gaussian distribution with small variance then $\mathbf{y}_i$ can be recovered by exploring the principal components of the data.

5.2.2 Recognizing nonlinear patterns from data

The set X_m may contain subtle data patterns and its distribution may be highly non-Gaussian. Hence to recognize nonlinear patterns, we need to take into account higher order correlations among the data. Nonlinear structure can be extracted by first mapping the input data into some feature space $\mathcal{F}$ via nonlinear transformation $\Phi : \mathbb{R}^d \rightarrow \mathcal{F}$ and then uncovering the low-dimensional subspace structure [98]. This maps X_m to $Z_m = [\Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2), \cdots, \Phi(\mathbf{x}_N)]$ and the covariance matrix C_Φ in $\mathcal{F}$ is given as,

$$C_\Phi = \frac{1}{N}\sum_{i=1}^{N} \Phi(\mathbf{x}_i)\Phi(\mathbf{x}_i)^T = \frac{1}{N} Z_m Z_m^T. \tag{5.3}$$

Consider the $N \times N$ dot product matrix $\mathcal{K} = \frac{1}{N} Z_m^T Z_m$ where

$$\mathcal{K}_{ij} = \frac{1}{N}\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) = \frac{1}{N}\kappa(\mathbf{x}_i, \mathbf{x}_j). \tag{5.4}$$

Here $\kappa(\mathbf{x}, \mathbf{y}) = exp(\frac{||\mathbf{x}-\mathbf{y}||}{2\sigma^2})$ is the Gaussian kernel function. Let λ_i be the eigenvalue of $\mathcal{K}$ corresponding to eigenvector $\mathbf{u}_i = [u_{i,1}, u_{i,2}, \cdots, u_{i,N}]^T$. C_Φ also has the same eigenvalues with one-to-one correspondence between nonzero eigenvectors of $\mathcal{K}$ and the nonzero eigenvectors ($\mathbf{v}_i$) of C_Φ. So we can write the relation, $\mathbf{v}_i = \alpha Z \mathbf{u}_i$ where α is normalization constant. If both the eigenvectors have unit length then $\alpha = \frac{1}{\sqrt{\lambda_i N}}$. Let $\|\mathbf{u}_i\| = \frac{1}{\sqrt{\lambda_i N}}$ so that $\alpha = 1$. For the data sample $\mathbf{x}_i$, its low dimensional j^{th} principal component can be computed using kernel function as,

$$y_{ij} = \mathbf{v}_i \cdot \Phi(\mathbf{x}_i) = \sum_{j=1}^{N} u_{i,j}\ \kappa(\mathbf{x}_i, \mathbf{x}_j). \tag{5.5}$$

Here, the nonlinear principal component (H_n) is defined similarly as (5.2), except the fact that only the role of $\mathbf{u}_i$ and λ_i are changed according to the present context.

5.2.3 Algorithm

Multiclass classification problem can be reduced to a set of binary classification problems, where data points from one particular class are seen as positive examples (minority class), while those outside the class are considered to be negative examples. This is 'one-vs-rest' classification procedure, where the 'rest' or the negative examples form the majority class.

Let $X = \{X_M \cup X_m\}$ be the entire dataset having $(M + N)$ data-points and $\mathbf{x}_i \in X$ where $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \cdots, x_{i,d}]^T$. Majority class set X_M contains (d-dimensional) M feature points where as minority class X_m contains N feature points. For balancing the majority and minority class size, we need to synthesize N_o extra samples from the (unknown) data distribution that might have generated the given minority class $X_m = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^N$. Since this distribution is completely unknown, we have to take help from the recognized linear (H_l) and nonlinear (H_n) patterns of the minority class. Each new synthetic sample $\hat{\mathbf{x}}_i$, is generated within k-nearest neighbor locality of the current sample $\mathbf{x}_i$. Locality sampling prevents the original inherent structure of X_m from collapsing. Before random imputation, H_l (or H_n) provides an important orientation information, about which direction to choose in the feature space for synthesizing $\hat{\mathbf{x}}_i$. This partially guided random oversampling helps to retain the intrinsic class properties and $\hat{\mathbf{x}}_i$ also adheres to subtle hidden patterns of the original class.

Algorithm 5.2.1, describes this partially guided random oversampling procedure. It is improvement to SMOTE [95], by introducing the proposed partial guiding mechanism (before random imputation stage) for synthesizing $\hat{\mathbf{x}}_i$. If the number of oversamples required N_o, is greater than total minority class samples N, then using each $\mathbf{x}_i$ we need to generate extra R (see step 5 in algorithm 5.2.1) synthetic samples. Otherwise only random percent of X_m will be G-SMOTEd

(see step 3). Steps 9-13 are laid to find $\mathbf{x}_n$, which is the neatest neighbor of $\mathbf{x}_i$ for which $\mathbf{p}_n$ (the vector joining $\mathbf{x}_i$ and $\mathbf{x}_n$) has the minimal absolute deviation (θ_n) in its orientation from H_l (or H_n). Thereby $\mathbf{x}_n$ serves as a beacon for synthesizing $\hat{\mathbf{x}}_i$. These guiding steps prevent overgeneralization of the minority class by conserving intrinsic class properties. Steps 14-18 generate a synthetic sample by random imputation along each attribute $x_{i,j}$ of the sample $\mathbf{x}_i$, which is proportional to the corresponding attribute difference ('*diff*' in step 15) from $\mathbf{x}_n$. The constant of proportionality is independently chosen (for each attribute) from uniform distribution $\mathcal{U}(0, 1)$. This randomness in the imputation helps to avoid acute specialization of minority class and thereby preventing the possible overfitting of classifier towards newly synthesized minority class.

After rebalancing the minority classes using G-SMOTE, final classification is performed by employing the bootstrapping and ensemble classifier. This procedure is described by the algorithm 5.2.2. Many times imbalanced dataset is skewed greatly, where $M >> N$. So it is not good idea to perform too much synthetic oversampling of the minority class. Because it may lead to over amplification of usual (unimportant) patterns and suppression of rare (important) properties of the minority class and thereby changing its overall intrinsic structure. Class imbalance problem generally arises when the ratio $\frac{M}{N}$ is greater than three (see step 1 in Algo.5.2.2) and the use of ensemble of classifiers is a good way to tackle it [92], [99]. Performing only undersampling of the majority class while keeping the minority class untouched, will not solve the class imbalance problem [91]. Firstly, because undersampling may discard potentially important samples from majority class and secondly, as minority class strength is small, during classifier learning phase it gets overshadowed by majority class and its hidden patterns remain unexpressed. Hence appropriate amplification of minority class samples is necessary. To solve these issues, we have employed bootstrapping by simulta-

Algorithm 5.2.1 G-SMOTE: partially Guided oversampling

input: Minority class X_m with $|X_m| = N$, number of nearest neighbors k, number of oversamples required N_o

output: Synthetic minority class samples $\{\hat{\mathbf{x}}_i \in \mathbb{R}^d\}_{i=1}^{N_o}$

1: amount of synthesis, $fraction = \frac{N_o}{N}$
2: **if** $fraction < 1$ **then**
3: randomize the X_m class samples
4: **end if**
5: synthesis runs per sample, $R = floor(fraction)$
6: get H_l and H_n from X_m
7: * *start generating each new sample* *\
8: **for** sample index i from 1 to N_o **do**
9: $\mathbb{K} \leftarrow$ set of k-nearest neighbors of $\mathbf{x}_i$
10: $\mathbb{P} = \cup_{n=1}^{k}\{\mathbf{p}_n\}$, a set of vectors joining $\mathbf{x}_i$ and $\mathbb{K}$
11: **for** synthesis runs r from 1 to R **do**
12: $\theta_n = cos^{-1}\left(\frac{|\mathbf{p}_n^T H_l|}{\|\mathbf{p}_n\| \cdot \|H_l\|}\right)$, $\forall \mathbf{p}_n \in \mathbb{P}$, (or use H_n)
13: $n = \text{argmin}_n\ \theta_n$
14: **for** each attribute j from 1 to d **do**
15: attribute difference, $diff = x_{n,j} - x_{i,j}$
16: random imputation, $rand = \mathcal{U}(0, 1)$
17: generation, $\hat{x}_{i,j} \leftarrow x_{i,j} + diff \cdot rand$
18: **end for**
19: **end for**
20: **end for**

neous oversampling of the minority class and undersampling of the majority class to build the ensemble of classifiers. To make sure that no sample from majority class is discarded, we have to perform exhaustive undersampling. Which finally leads to division of majority class into several (s) disjoint sets $\{X_i\}_{i=1}^{s}$ (see step 2). Over synthesis of the minority class samples may increase the possibility of classifier overfitting and inadequate synthesis may not offer any benefit. So only sufficient amount of samples (N_o) are synthesized using G-SMOTE (steps 3, 4) to form the new balanced minority class $\breve{X}$ (step 5). Then set $\breve{X}$ is collaborated with each X_i to train the set of s classifiers (step 6). For classifying a new query, final decision is made by majority voting of all classifiers in the ensemble.

Algorithm 5.2.2 Bootstrapping and ensemble classifier (EnC)

input: Majority class X_M, minority class X_m with $|X_m| = N$, $|X_M| = M$, $M >> N$, number of nearest neighbors k

output: Ensemble classifier

1: number of sets, $s = ceil(\frac{M}{3N})$

2: undersample X_M to generate s different disjoint sets, $X_M = \cup_{i=1}^{s} X_i$ such that $\forall\, i\ |X_i| > 3N$

3: total synthetic samples to generate, $N_o = \min_{\forall\, i} |X_i| - N$

4: oversample X_m to get $X_o = \textit{G-SMOTE}(X_m, N_o, k)$

5: class balancing by synthesizing, $\breve{X} = X_m \cup X_o$

6: get s classifiers by augmenting the data after balancing, $\mathbb{C}_i = classifier(X_i, \breve{X}),\ \forall\, i$

7: $\backslash *$ *make the decision based on majority voting* $* \backslash$

8: EnClassify $= decision(\mathbb{C}_1, \cdots, \mathbb{C}_s)$

5.3 Experimental Results

Multiclass classification problem can be simplified to two-class (by 'one-vs-rest') problem. Minority class labels are considered to be positive while that of majority class as negative. True positives (TP) and true negatives (TN) are the number of correctly classified positive and negative examples respectively. Similarly false negatives (FN) and false positives (FP) are from misclassification. For measur-

Table 5.1: Summary of the different imbalanced datasets used for the performance evaluation of the proposed method

Data sets	#of minority samples (N)	#of majority samples (M)	#of features (nominal/continuous)	Ratio of imbalance
SCOP	118	1263	147 (cts)	1:10.7
Satimage	415	5809	36 (cts)	1:14.0
HypoThy	131	2930	16 (nom), 8 (cts)	1:22.4
EuThy	151	3012	18 (nom), 7 (cts)	1:19.9

Table 5.2: Classification performance evaluation (AUC values) and comparison with various methods (EnC is Algo.5.2.2)

Datasets	Classifiers	Without EnC	SMOTE	SMOTE+EnC	Ours(H_l)	Ours(H_n)
SCOP	LogReg (LA)	0.761	0.821	0.826	**0.832**	0.829
	LogReg (SW)	0.629	0.782	0.828	0.836	**0.838**
	LogReg (NW)	0.687	0.798	**0.860**	0.858	0.846
Satimage	1NN	0.826	0.841	0.895	0.913	**0.918**
	SVM	0.958	0.960	0.963	0.966	**0.976**
	LogReg	0.841	0.848	**0.853**	0.849	0.851
HypoThy	1NN	0.819	0.825	0.841	0.871	**0.910**
	SVM	0.892	0.895	0.901	**0.921**	0.913
	GaussProcess	0.944	0.949	0.951	0.971	**0.973**
EuThy	1NN	0.691	0.691	0.693	0.708	**0.712**
	SVM	0.921	0.926	0.929	0.938	**0.940**
	GaussProcess	0.931	0.938	0.947	**0.952**	**0.952**

ing performance of the learning systems, *error rate* = (FP+FN)/total samples, and *accuracy* = (TP+TN)/total samples, measures are not appropriate when the prior class probabilities are very different. Because these measures do not consider misclassification costs, are sensitive to class skewness and are greatly biased to favor the majority class [100], [101]. For skewed datasets with unequal error costs, as in biomedical analysis [28], it is more appropriate to use the Receiver Operating Characteristic (ROC) analysis [95]. For overall performance assessment, the useful measure is the fraction of total area that falls under ROC curve (AUC) [102]. The larger the AUC value, the better the classifier performance.

In this work, the proposed method is applied to 4 different imbalanced datasets. Table 5.1, shows their detailed description. From the class imbalance ratio, it is evident that minority class samples are extremely rare. These datasets include, structural classification of proteins (SCOP) database (Accession number: PCB00019) [103] which is divided in to 55 classification tasks, multispectral satellite images (Satimage) for soil type classification and hypothyroid

(HypoThy), sick euthyroid (EuThy) [104] are two medical diagnoses datasets. In SCOP dataset, the pairwise sequence similarity was employed to convert protein sequences into feature vectors. Three methods, namely local alignment (LA) kernel [105], Smith-Waterman (SW) [106] and Needleman-Wunsch (NW) algorithm [107] were adopted to measure the similarity between a pair of protein sequences and thereby constructing different feature spaces. LA kernel measures the similarity between a pair of protein sequences by constructing a kernel function, while SW is the local and NW is the global sequence alignment algorithm.

A series of widely used pattern classifiers were employed to demonstrate the performance of the proposed method for rebalancing the skewed datasets. They include, the nearest neighbor classifier (1NN), Support Vector Machines (SVM), logistic regression (LogReg) and Gaussian process for binary classification (GaussProcess) [108]. For appropriately assessing the benefits of the proposed framework, these classifiers were operated to their default parameters and those were kept constant throughout the experimentation.

Table 5.2, shows the detailed classification performance comparison according to AUC values and comparison with state-of-the-art minority oversampling method SMOTE [95]. SCOP has 55 different classification tasks, so for computational advantage LogReg classifier was used in 3 different feature spaces and the results were averaged over all classification tasks. The column 'Without EnClassifier' shows the classification performance for the original imbalanced data, where none of the rebalancing techniques were used. When the SMOTE was accompanied by the algorithm 5.2.2 by replacing its step 4 with SMOTE (SMOTE+EnClassifier), the performance improved a little more. This shows the advantage of the bootstrapping with simultaneous oversampling of the minority class and undersampling of the majority class to create the ensemble of classifiers. The last two columns show performance of the overall proposed data rebalanc-

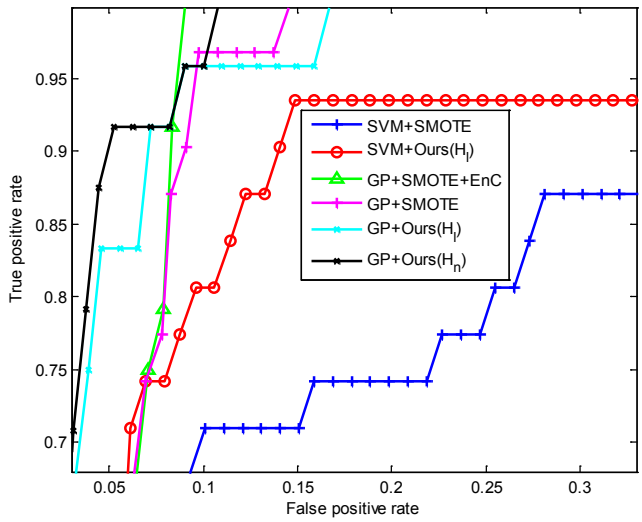


Figure 5.1: Receiver operating characteristic (ROC) curves: Classification performance evaluation for *HypoThy* dataset with SVM and Gaussian process (GP) binary classifiers using amalgamation of various dataset rebalancing methods. EnC is the ensemble classifier (algorithm 5.2.2). (Best seen in color)

ing technique, with the help of extracted linear (H_l in sec. 5.2.1) and nonlinear (H_n in sec. 5.2.2) patterns from the minority class. It shows that the proposed dataset rebalancing technique outperforms the other methods in many classification tasks. The column 'Ours with (H_l)' shows the fact that, for the datasets having compact class representation (low intra-class variations), viz., HypoThy and EuThy, the extracted linear patterns (H_l) also serve as a good guidance for performing the hybrid sampling. After rebalancing the minority class, the performance has increased across all the classifiers. Because it prevents classifiers from getting biased towards majority class and thereby classifiers can draw the clear class separating boundaries in the feature space. Difference between performance of the 'Without EnClassifier' and the dataset rebalancing methods, shows importance of the solving class imbalance issue before recognizing the patterns from it.

Fig. 5.1 shows the classification performance evaluation of HypoThy dataset in terms of detail ROC curves. For better visual analysis and clear discrimination, only few ROC curves are plotted and note scales of the axes. The performances of SVM and GP classifiers without using any dataset rebalancing technique were very low, so they are not shown here (more details are given in Table 5.2). The proposed oversampling method guided by linear patterns (H_l) improved the classification performance using SVM classifier as compared to the existing SMOTE [95]. Though performance was not the highest in this case (analysis via SVM classifier), it showed the importance of proper dataset rebalancing technique in terms of relative performance improvement. When same dataset was rebalanced with the proposed method using nonlinear patterns (H_n), the Gaussian process classifier (GP) gave the highest classification AUC. When SMOTE was accompanied by ensemble classifier, the overall framework has further boosted the performance.

5.4 Discussion

Strategic oversampling of the minority class is an important solution to the class imbalance problem, rather than completely random oversampling or sampling with replacement (i.e. oversampling by replication). Because generating synthetic samples randomly within a minority class, will not adhere to the original class properties. Thus newly generated samples will not follow the probability distribution from which the original minority samples might have been generated. On the other hand, those synthesized extra samples were then put back in the original minority class for balancing (enlarging) the class size and it adversely affects the original class structure. Then though the class size balances, the newly synthesized class no longer closely resembles the original minority class. The reason for under-performance of the sampling with replacement can be under-

stood by analyzing in the feature space. As the minority samples get replicated, the learning algorithm becomes more confident about the decision region in the feature space and it becomes more and more specific to the minority class. This specialization generally leads to further shrinkage of the decision boundary and the learner carves it very closely to the minority class region. That is specialization hinders the generalization. Without any type of oversampling, the learning algorithm will be given extremely low minority class samples and so it gets biased towards the majority class in the classification tasks. Hence imbalanced datasets must be handled appropriately via rebalancing strategy (like G-SMOTE Algo.5.2.1) before handing them over to the pattern classifiers.

5.5 Conclusion

In this work, we propose the complete framework for handling the imbalanced dataset for pattern recognition. We have employed bootstrapping by simultaneous oversampling of the minority class and the undersampling of the majority class to build the ensemble of classifiers. To perform oversampling, first linear and nonlinear patterns are extracted by considering the second order and higher order correlations respectively within the minority class. Then these patterns are used to partially guide the random imputation in each dimension of the feature space to generate the synthetic sample. This guided randomization helps to preserve the minority class properties and unlike the existing synthetic minority oversampling technique (SMOTE) it prevents over-generalization of the minority class. Exhaustive undersampling of the majority class ensures that none of its important samples get neglected. The ensemble classifier helps further to boost the classification performance.

Chapter 6

Traffic Activities: Features and Mulit-task Learning

6.1 Introduction

Uncovering the hidden motion patterns in the traffic scene is an essential part of the intelligent visual surveillance systems. Intersections are the important parts of the road networks, especially for traffic surveillance because of their high variable structure, presence of multiple flows of the vehicles, abrupt motions and the mixed traffic that ranges from pedestrians to bicycles and up to the heavy vehicles. Monitoring these variety of traffic events requires high level understanding of the scene. Traffic rules govern the behavior of the moving objects, which give rise to typical motion patterns in the scene. Hence grouping of similar motions patterns and training the classifiers for their modeling are crucial for scene understanding and detecting the abnormal motion patterns in the surveillance videos.

A number of approaches have been proposed for motion features extraction such as using optical flow [9, 109, 110] and its statistical features [111, 18], SIFT

Figure 6.1: Typical motion patterns in distinct traffic scenes. They have different geometric properties (size, location, orientation) but they share the similar semantic content (right-turn, linear motion).

flow [112], active region selection procedure [113] or trajectories [114]. These features are clustered into multiple groups with each group is representing a typical motion pattern. For modeling these motion patterns, a diversity of methods have been proposed by using sparse coding [115], mixture of Gaussian [116], dynamic textures [117] and probabilistic topic models [109, 110, 9]. But existing works generally assume that sufficient amount of training data is available for modeling the traffic patterns, so performance of these methods is usually not guaranteed without enough data. However, in surveillance applications, it is time consuming and expensive to collect sufficiently large amount of labeled data from different viewpoints and all possible traffic scenes. Hence, discovering hidden motion patterns in traffic scene is a challenging problem with a great practical importance.

The existing direct classification approach is likely to have unreliable performance because of data scarcity and its imperfect, noisy nature. Such explicit systems, using rules adjusted according to each traffic scene by trial and error, rarely provide generic and robust solutions, especially when the environment and motion patterns change. So better solution is to design a more generalized model that can learn from even less amount of data. Some motion patterns may be shared in several traffic scenes as shown in Fig. 6.1 (yellow arrow). So prior knowledge obtained from certain scenes (source view) can be employed for modeling motion patterns in another scene (target view) with insufficient data. The

approach of utilizing prior knowledge from source domain to improve the classifier in target domain, where the training samples are not sufficient, is called the transfer learning [118, 119]. The approaches like domain adaptation [120], sentiment classification [121] and adaptive transfer learning [118] handle the target domain data insufficiency problem by leveraging the prior knowledge obtained from source domain. Though these approaches can address data scarcity issue, the main problem with them is that the feature distribution in target domain might have quite different statistical characteristics as compared to source domain. For example, the *right-turn* pattern (yellow arrow) in Fig. 6.1 (c) is relatively different from that of (a) and (b). They make implicit assumption of availability of the generalized source domain and the sufficient training samples, but which does not hold for most of the practical surveillance scenarios.

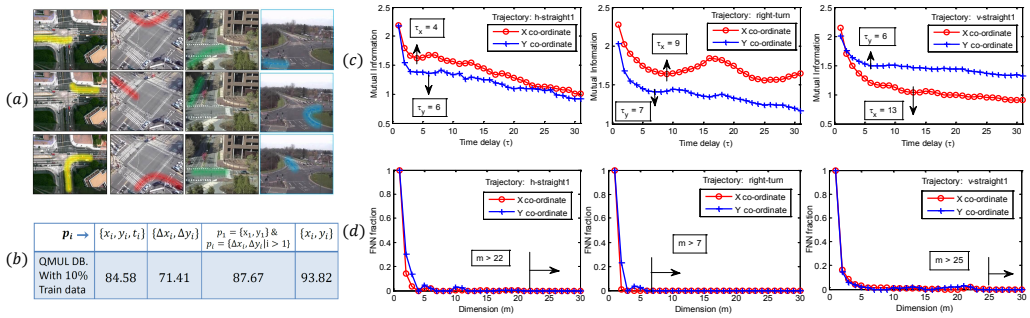
So to solve these problems we do not use any source domain information. Instead we make use of the fact that motion patterns can also share semantic information among different patterns within the same traffic scene, like the *straight-line* patterns (green arrow) in Fig. 6.1 (a) share the similar underlying linear dynamics. To share the semantic information, we employ multi-task joint feature learning scheme [122, 123], where all the pattern classifiers are trained simultaneously by exploiting the correlations among different motion patterns. Training data scarcity problem is tackled through strategic oversampling of the raw trajectories along with high dimensional embedding of the time series data.

6.2 Feature construction

Important clues for uncovering hidden motion patterns in the surveillance video are obtained by answering, 'where' and 'how' the activity has occurred over the course of time. The answer to the first question reveals the spatial distribution

of the activity occurrences, whereas the answer to the second one discovers the underlying motion dynamics. The features should encapsulate these answers in order to extract vital information from the video that is necessary for recognition of the motion patterns from the video.

In traffic scenario, detailed spatio-temporal motion information can be represented in terms of trajectories, which can be easily and reliably extracted by using existing methods based on the blob-based tracking (using foreground estimation or vehicle detection) or the low-level feature tracking (using SIFT, Harris corner interest points) [124, 125, 126, 127]. So recognition of motion patterns in a traffic scene is nothing but classification of the corresponding trajectories into appropriate groups. A trajectory is composed of a set of 3-dimensional interest points (x_i, y_i, t_i), where the consistent motion has occurred over the course of time [118]. These trajectories are categorized and labeled by employing latent Dirichlet allocation [128] to obtain the training data. Vehicle dynamics is captured in every video frame, so time dimension (t_i) is redundant in trajectory description. Different types of trajectory descriptors are experimentally evaluated and compared in Fig.6.2(b). From those we simply use the spatial interest points, $p_i = [x_i, y_i]$, as the trajectory descriptor, which is compact, computationally fast and producing high accuracy compared to other variants. Temporal information is embedded implicitly as these interest points are consecutively arranged one after another. So the descriptor $\mathcal{Z}^c$ which represents a single trajectory from class $c \in (1, \cdots, C)$, is a concatenation of M interest points, $\mathcal{Z}^c = \{p_1, p_2, \cdots, p_M\}$. It can not be used as a feature directly since it has different dimensionality depending upon M. So we have to construct the equidimensional informative trajectory feature $(\mathbf{z}^c)$ from the raw $\mathcal{Z}$s. In addition to this, the number of per class training samples is far less to achieve good generalization of the classifier. We address these issues in the following section by proposing trajectory construction strategy to get same



$p_i \rightarrow$	$\{x_i, y_i, t_i\}$	$\{\Delta x_i, \Delta y_i\}$	$p_i = \{x_i, y_i\}$ & $p_i = \{\Delta x_i, \Delta y_i \mid i > 1\}$	$\{x_i, y_i\}$
QMUL DB. With 10% Train data	84.58	71.41	87.67	93.82

Figure 6.2: (a) Few exemplar motion patterns in 4 datasets viz. NGSIM (1^{st} column), WI (2^{nd}), MIT (3^{rd}) and QMUL (4^{th} col.); (b) Different trajectory descriptors and their performance on QMUL database. As $p_i = \{x_i, y_i\}$ has performed well, we used it for further analysis; (c) & (d) Embedding delay pair (τ_x, τ_y) and dimension (m) estimation for 3 different patterns from QMUL.

dimensional feature vector $\mathbf{z}^c$ and efficient trajectory over-sampling method to generate enough number of appropriate training samples.

6.2.1 Trajectory construction and over-sampling

Each raw trajectory descriptor $\mathcal{Z}^c$ is analyzed in to separate time series data, like $\mathcal{X}^c = (x_1, x_2, \cdots, x_M)$ and $\mathcal{Y}^c = (y_1, y_2, \cdots, y_M)$. Embedding is defined as the mapping from 1-dimensional space to m-dimensional space. It is achieved by Taken's theorem [129], which states that a map exists between the original space and the reconstructed space after embedding. According to it, the dynamical properties of the system in the true state space are preserved under the embedding transformation. The key idea behind embedding is that all the variables of the dynamical system are generically connected, i.e. they influence one another. For a given one dimensional time series $\mathcal{X}^c$, every subsequent point, x_i, results from an intricate combination of the influences of rest of the variables. Hence, $x_{i+\tau}$ may be viewed as the second substitute system variable which carries information about other variables within time interval τ. With the same reasoning we can introduce

a series of substitute variables $x_{i+2\tau}, \cdots, x_{i+m\tau}$, which carry the same information as the original system variables [130]. For a large enough embedding dimension m, the delay vectors $\mathbf{x}_i(\tau) = [x_i, x_{i+\tau}, \cdots, x_{i+(m-1)\tau}]$ with $i \in (1, \cdots, \tau)$, generate an embedded phase space that has exactly the same properties as that formed by the original system variables. However, Taken's theorem does not provide a method to find the optimal values of the embedding parameters, τ and m, which we estimate here by using the mutual information [131] and the false nearest neighbor (FNN) algorithms [132]. Then we perform the trajectory oversampling, which is given in algorithm 6.3.1.

Estimating embedding delay (τ)

The mutual information between x_i and $x_{i+\tau}$ can be used to estimate proper τ. The value of τ should be large enough such that x_i is measuring something significantly different from $x_{i+\tau}$, which helps to retain new unseen information. But τ should not be larger than the time in which system loses its initial state memory. For a given time series $\mathcal{X}^c$, we partition $d = |min(\mathcal{X}^c) - max(\mathcal{X}^c)|$ into B equally sized bins. If P_h and P_k denote the probabilities that variable takes a value inside h^{th} and k^{th} bin, and $P_{h,k}$ is the joint probability that x_i is in bin h and $x_{i+\tau}$ is in bin k, then the mutual information with various delays is computed as,

$$I(\tau) = -\sum_{h=1}^{B}\sum_{k=1}^{B} P_{h,k}(\tau)\, ln\frac{P_{h,k}(\tau)}{P_h(\tau)P_k(\tau)}. \tag{6.1}$$

Then embedding delay is given as, $\tau^* = \arg(1^{\text{st}})\min_{\tau}\ I(\tau)$.

Estimating embedding dimension (m)

The optimal embedding dimension is estimated by using the algorithm given in [132]. It assumes that there are no sudden irregularities in the system dy-

namics, which is realistic in in our traffic scenario where the constraints of traffic rules make the state space of dynamical system folds and unfolds smoothly. This translates to the observation that the nearest neighbors should remain close to each other during forward embedding iterations. If the neighbor of a point fails to satisfy this criterion then it is termed as false nearest neighbor (FNN) and the algorithm chooses m that minimizes FNNs as follows. Given a point $\mathbf{x}_i(\tau)$ in m-dimensional space from $\mathcal{X}^c$ and we find its neighbor $\mathbf{x}_j(\tau)$, so that $\|\mathbf{x}_i(\tau)-\mathbf{x}_j(\tau)\| < \xi$. Then normalized distance between D_i between the $(m+1)$th embedding coordinate of those points is computed as, $D_i = \frac{x_{i+m\tau}-x_{j+m\tau}}{\|\mathbf{x}_i(\tau)-\mathbf{x}_j(\tau)\|}$. If D_i is greater than given threshold then $\mathbf{x}_i(\tau)$ is marked as having a FNN. D_i is found for entire $\mathcal{X}^c$ and for various $m = 1, 2, \cdots$ until the fraction of FNNs is negligible to obtain an optimal embedding dimension m_x^c. Using similar procedure m_y^c for $\mathcal{Y}^c$ is found. Final embedding dimension for a given dataset is chosen as, $m^* = \max_{c\in(1,\cdots,C)}\{m_x^c, m_y^c\}$.

6.3 Classification by joint feature selection

Different class of traffic patterns share some common information as they are correlated with each other. For example, horizontal straight and vertical straight movements are two different motion patterns but they share the common underlying dynamics of spatial linear motion. We have C-class classification problem. We convert this into C one-versus-rest binary classification tasks. As these learning tasks are not completely independent, so it is not fruitful to separately train each binary classifier. Because, learning of one task may be helpful for the learning of other tasks. In Multi-task learning (MTL) approach, multiple related tasks are learned in parallel with some shared representation [122]. This greatly improves the learning efficiency, as each task also learns from other related task and

thereby achieving a good generalization. One way to capture the task relatedness from multiple tasks is to constrain all classifier models to share a common set of features. This motivates the group sparsity regularized simultaneous training of the models [123].

Let $\mathbb{D}^c = \{(\mathbf{z}_i^c, l_i^c) | \, i = 1, \cdots, n^c\}$ be the available training data for the binary classification task $c \in (1, \cdots, C)$, where $\mathbf{z}_i^c \in \mathbb{R}^{2m}$ is the feature sample with label $l_i^c \in \{1, -1\}$ corresponding to either 'class c' or 'not class c' (one-vs-rest classification strategy). Let the parameters $\mathbf{w}^c$ and b^c define the model for task c. Define a matrix $W = [w^1, \cdots, w^C]$ and a vector $\mathbf{b} = [b^1, \cdots, b^C]^T$. With employing joint sparsity and model complexity regularization the multi-task learning framework is given as follows,

$$\min_{W,\mathbf{b}} \sum_{c=1}^{C} \sum_{i=1}^{n^c} \mathcal{L}(\mathbb{D}^c; \mathbf{w}^c, b^c) + \lambda_1 \|W\|_{2,1} + \lambda_2 \|W\|_F^2, \tag{6.2}$$

where the hinge loss is defined as, $\mathcal{L}(\mathbb{D}^c; \mathbf{w}^c, b^c) = \max(0, 1 - l_i^c(\mathbf{w}^{cT}\mathbf{z}^c + b^c))$. The Frobenius norm is defined as $\|W\|_F^2 = \sum_c \|\mathbf{w}^\mathbf{c}\|_2^2$ and $\|W\|_{2,1} = \sum_{i=1}^{2m} \sqrt{\sum_c w_i^{c2}}$. The $\ell_{2,1}$-norm regularization encourages multiple predictors from different tasks to share similar parameter sparsity patterns. The regularization parameters λ_1 controls the amount of group sparsity and λ_2 controls the overall model complexity. This indeed helps in achieving good generalization performance. Loss function and norm regularizes are convex hence it is a convex but non-smooth optimization problem. We applied accelerated proximal gradient method [133, 123] to solve the optimization problem (6.2). The class label of the new input sample $\hat{\mathbf{z}}$ is found as,

$$c^* = \underset{c}{\operatorname{argmax}} (\hat{\mathbf{z}}^T \mathbf{w}^c + b^c). \tag{6.3}$$

6.4 Inference and classification

Given a raw query trajectory $\mathcal{Z}^q$, its class label q is what we have to infer. But as described in sec. 6.2.1, the embedding delay (τ^c) depends upon the trajectory class label c. So for performing oversampling we innumerate through all embedding delay pairs $\{\tau_x^c, \tau_y^c\}_{c=1}^{C}$. Oversamples ($\hat{\mathbf{z}}$ of $\mathcal{Z}^q$) obtained by using $\{\tau_x^c, \tau_y^c\}$ are stored in the set $\mathbb{Z}^c$ and are classified using (6.3) to produce the corresponding output label set $\mathbb{L}^c$. As $q \in (1, \cdots, C)$, the $\mathcal{Z}^q$ will get shattered appropriately according to true underlying delay pair in one of the sets $\mathbb{Z}^q \in \{\mathbb{Z}^c\}_{c=1}^{C}$, and corresponding $\mathbb{L}^q$ will have almost identical elements with value q. Since $\mathbb{L}^q$ has very less variability in its elements, its entropy is very low. Hence the class label q of the query trajectory is inferred as the most frequent element in the lowest entropy set from $\{\mathbb{L}^c\}_{c=1}^{C}$. Algorithm 6.4.1 describes this procedure.

6.5 Experimental results

6.5.1 Datasets and Evaluation details

We evaluated the proposed approach on four different traffic datasets, viz. QMUL junction dataset [9], MIT dataset [10], NGSIM dataset [11] and the Wide Intersection (WI) is our own dataset of traffic scene at eight-lane road. Each of these

Algorithm 6.3.1 Trajectory oversampling subroutine

input: raw trajectory $\mathcal{Z}^c$, m, embedding delay pair $\{\tau_x^c, \tau_y^c\}$
output: set of oversamples $\mathbb{Z}^c = \{\mathbf{z} \in \mathbb{R}^{2m}\}$

1: $\mathbb{Z} \leftarrow \emptyset$; $\{\mathcal{X}^c, \mathcal{Y}^c\} \leftarrow \mathcal{Z}^c$; $\quad \backslash * \textit{ separating } x, y \textit{ coordinates } * \backslash$
2: **for** each $\tau \in \{\tau_x^c, \tau_y^c\}$ and $i \in (1, \cdots, \frac{|\mathcal{Z}^c|}{2} - (m-1)\tau)$ **do**
3: $\quad \mathbf{z}_i = \left[x_i, y_i, \cdots, x_{i+(m-1)\tau}, y_{i+(m-1)\tau}\right], \quad x_i \in \mathcal{X}^c, y_i \in \mathcal{Y}^c$
4: $\quad \mathbb{Z} \leftarrow \mathbb{Z} \cup \mathbf{z}_i$
5: **end for**

Algorithm 6.4.1 Trajectory pattern recognition

input: query trajectory $\mathcal{Z}^q$, trained model $W, \mathbf{b}$, embedding delays $\mathbb{T} = \{\tau_x^c, \tau_y^c\}_{c=1}^{C}$, embedding dimension m

output: Recognized class label q

* *oversample considering each embedding delay* *\

for each class c from 1 to C **do**

  $\mathbb{Z}^c = oversample(\mathcal{Z}^q, m, \tau_x^c, \tau_y^c)$ * *algo. 6.3.1* *\

  $\mathbb{L}^c \leftarrow$ get the set of class labels $\forall\, \hat{\mathbf{z}} \in \mathbb{Z}$ using (6.3)

  $K \leftarrow$ # distinct elements in $\mathbb{L}^c$

  $\{n^k\}_{k=1}^{K} \leftarrow$ # elements in $\mathbb{L}^c$ having class label k

  $H(\mathbb{L}^c) = -\sum_{k=1}^{K} \frac{n^k}{|\mathbb{L}^c|} \cdot \log_2(\frac{n^k}{|\mathbb{L}^c|})$ * *entropy* *\

end for

$\mathbb{L}^* = \text{argmin}_{\mathbb{L}^c}\ H(\mathbb{L}^c)$

$q = mode(\mathbb{L}^*)$

consists of different types of motion patterns like linear trajectories at different locations and non-linear trajectories with different shapes. Sample frames from all the datasets are shown in Fig.6.2(a), which shows a wide range of variations in viewpoints, background noise and illumination changes. The detail motion patterns in QMUL dataset are delineated in Fig.6.3(a) with sample frames and by labels of the corresponding confusion matrix. The F-measure has been used for the performance evaluation, which is the harmonic mean of precision and recall. We have chosen training samples randomly and with varying amount (see table 6.1) for each experiment and the average results have been reported after repeating the experiments for five times. If the model is trained using only one trajectory from each type of the pattern then we call it 'One-shot' learning.

6.5.2 Performance analysis

We have defined the most commonly used SVM based classification [134] as the baseline method. For dealing with multiple classes, we employed 'one-vs-rest' binary classification strategy. Unlike the proposed oversampling technique, in

baseline method the trajectories were uniformly sampled to obtain equidimensional feature vector, whose dimension was equal to the length of the shortest trajectory in the dataset. We have compared the performance on all datasets with the existing adaptive transfer learning (a-TL) [118] method. It works with less number of training samples (in target domain) by making transfer of information (from source domain). We have used NGSIM as the source dataset for a-TL [118] method because it has been captured from top-view and has contained typical traffic patterns. Where as the proposed method handles the training data scarcity problem by strategic oversampling and multi-task learning framework, so it does not require any separate source dataset. We have followed the same experimental setup for performance comparison with different methods.

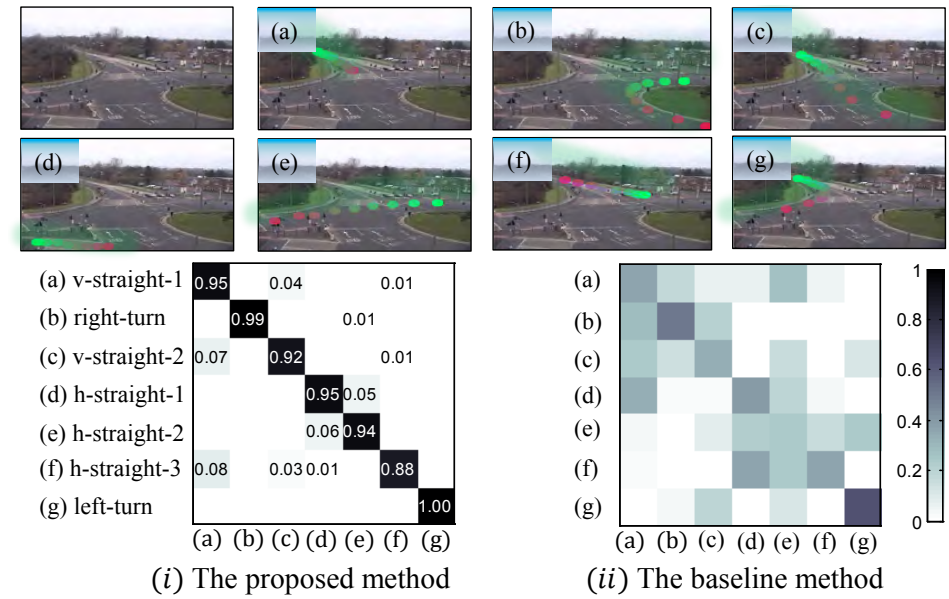
Embedding delay and dimension estimation. For feature construction, embedding delay (τ) is a quite important parameter which controls the amount of motion pattern information retained in the final fix dimensional (m^*) feature vector. So embedding delay pair $\{\tau_x^c, \tau_y^c\}$, was separately estimated for each class (c) of the motion pattern using the algorithm given in sec. 6.2.1. For preserving class specific information as well as achieving robustness to outliers and noise, we used average of the $I(\tau)$ vs τ curves using all $\mathcal{Z}^c$s from the same class. Fig.6.2(c) and (d) show the embedding delay pair and feature dimension estimation results respectively for three types of motion patterns from QMUL dataset, which are shown in the last column of Fig.6.2(a).

Average precision. Table 6.1, shows the average precision obtained by different methods on all the datasets after using different amounts of training data (from 10 to 75%). The fewer the training samples, the worse the performance of the baseline method across all datasets. Moreover it fails to perform one-shot learning. On the other hand, although a-TL achieves higher performance (even

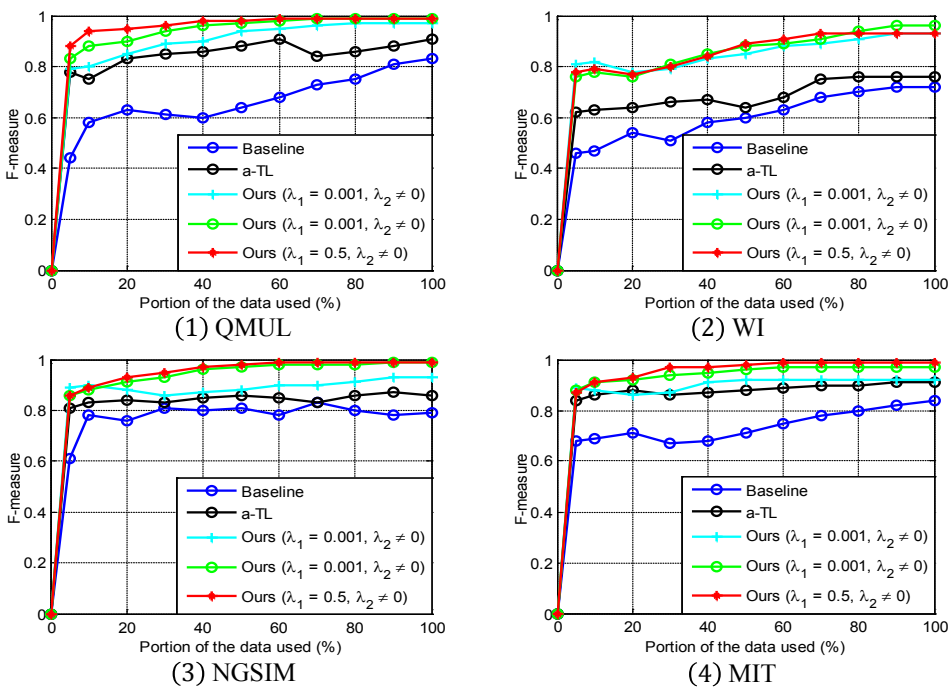
in case of less training data) than baseline using transfer of information from source domain, improvement is not great as compared to the proposed method. Additionally it requires prior source domain data. This means that the proposed framework makes a significant contribution in performance improvement by using just the target domain scarce data. Fairly good performance of the proposed method even using a single training example (one-shot learning) can be attributed to the efficient oversampling strategy which amplifies the hidden motion pattern and the multi-task learning framework which learns all the models simultaneously. This indicates the importance of handling data scarcity issue in the classifier model itself, otherwise the model will not generalize appropriately.

Joint feature selection benefit. When we set the group sparsity regularization parameter to zero ($\lambda_1 = 0, \lambda_2 \neq 0$) in optimization problem (6.2), we were hindering the joint feature selection procedure and thereby preventing the multi-task learning (see table 6.1). In this case a model corresponding to each motion pattern was learned independently and $\lambda_2 \neq 0$ helped to prevent the overfitting. As all the models were not simultaneously trained, they could not make use of correlation among the different motion patterns. Thus there was no information sharing or transfer among different models during their learning. Hence the precision corresponding to method 'Ours ($\lambda_1 = 0$)' was lower than the complete framework given in (6.2). But the higher performance of 'Ours ($\lambda_1 = 0$)' than baseline could be accredited to the proposed efficient oversampling strategy and the generalization achieved by model complexity regularization ($\lambda_2 \neq 0$). After employing joint feature selection strategy (Ours ($\lambda_2 = 0$)) the performance improved a lot but it was not the highest because lack of model complexity regularization. The performance assessment from Table 6.1 shows benefits of the joint feature selection strategy.

Incremental learning. Throughout this experimental setting, test samples (40% from each class) were kept fixed and the remaining portion of the data (training) was fed gradually to the classifiers to see the incremental learning capability. Fig. 6.3(b), shows the graphs of F-measure versus the portion of training data used for different algorithms



(a)



(b)

Figure 6.3: (a) QMUL dataset typical motion patterns (top rows) and their recognition rate. (b) F-measure vs training data curves on four traffic datasets.

Table 6.1: Performance after using varying portion of training data

Average precision in percentage						
Dataset	Different Methods	One-shot	10%	25%	50%	75%
QMUL	Ours	41.44	**93.82**	**95.01**	**96.53**	**97.39**
	Ours (λ_2=0)	**41.52**	92.27	94.81	95.86	96.60
	Ours (λ_1=0)	40.21	89.22	90.89	93.31	94.18
	a-TL [118]	32.70	75.43	82.80	87.17	90.24
	Baseline [134]	-	51.85	67.47	70.88	78.21
WI	Ours	**39.33**	**89.70**	**94.26**	94.73	**95.76**
	Ours (λ_2=0)	38.09	88.53	92.03	**95.21**	95.69
	Ours (λ_1=0)	37.41	85.14	91.86	93.04	94.55
	a-TL [118]	18.32	63.01	69.23	82.10	85.47
	Baseline [134]	-	53.27	63.07	79.70	81.54
NGSIM	Ours	28.40	**91.10**	**94.80**	**98.48**	**98.31**
	Ours (λ_2=0)	**28.79**	89.63	93.07	97.81	97.91
	Ours (λ_1=0)	28.02	88.96	91.21	97.22	97.52
	a-TL [118]	13.87	81.75	83.17	87.89	88.23
	Baseline [134]	-	76.20	81.34	80.57	87.19
MIT	Ours	**34.11**	**90.39**	**93.18**	**95.72**	**97.37**
	Ours (λ_2=0)	34.03	88.98	93.06	94.97	96.76
	Ours (λ_1=0)	33.12	84.83	91.71	92.88	94.09
	a-TL [118]	12.46	84.87	91.00	88.96	90.61
	Baseline [134]	-	61.67	68.91	72.67	84.87

across all datasets. As the training data prospered, the performance of all the algorithms gradually increased. But a-TL and baseline algorithms did not achieve a satisfactory performance even after using a large amount of training samples. The proposed method quickly achieved the highest performance as training data portion was slightly increased. Here $\lambda_1 = 0.001$ prevents the model from fully learning by using correlations among the tasks, so it performed poorer than the one with $\lambda_1 = 0.5$.

Discrimination capability. Fig. 6.3(a), shows seven typical motion patterns in QMUL traffic scenario and the confusion matrix for the traffic pattern classification result. We selected 10% of the samples from each pattern for training the classifiers. The confusion matrix with dominant diagonal terms (see Fig.6.3($a \rightarrow i$)) (The brighter the color, the higher the classification result), illustrates the discriminability of the proposed method. The baseline method (see Fig.6.3($a \rightarrow ii$)) confuses a lot among the patterns having similar semantic meaning.

6.6 Conclusions

Here we presented a novel approach for motion pattern recognition in the traffic scenes, by employing an efficient trajectory oversampling strategy, and training all the pattern classifiers simultaneously with joint feature selection strategy. The time series embedding procedure helps to construct the equidimensional compact features as well as it shows a way for an efficient oversampling procedure from the raw trajectory data. Training data scarcity problem is tackled by amplifying subtle motion patterns hidden in the raw trajectories using strategic oversampling and making use of correlations among different patterns via muli-task learning framework. Experimental results show that the constructed features provide a natural, compact and discriminative representation for reciprocating motion patterns. This approach is computationally efficient with an advantage of representing detailed spatio-temporal motion information, unlike other methods e.g. bag-of-words approach which ignores the spatio-temporal dependencies.

Chapter 7

Audio Activities Recognition

7.1 Introduction

Audio event recognition (AER) or classification is a sub-area of auditory scene analysis that deals with the automatic understanding of audio data without human efforts. This area has received increased attention in the research community in recent years. This is mainly because, apart from its straightforward applications in audio retrieval, indexing and automatic tagging, it has the potential to play a pivotal role in perceptually aware interfaces such as computer or robotic assistance in meeting-room, video-surveillance, industrial automation and process control or mobile robots working in diverse environments [135]. For example in a video surveillance system, the use of audio information along with the video can enhance the performance of the system for scene understanding especially in dark illumination conditions or in areas outside the camera view, otherwise the underlying activity recognition using only video will become impossible due to absence of perceptual information. In case of a meeting-room assistant robot, detection or classification of certain audio events related to human presence like *chair moving*, *door open*, *keyboard typing* etc. may help the robot to detect and recognize the activities occurring in the room. In all these scenarios, the performance of the entire control and automation system then highly depends on its ability to correctly classify an audio event.

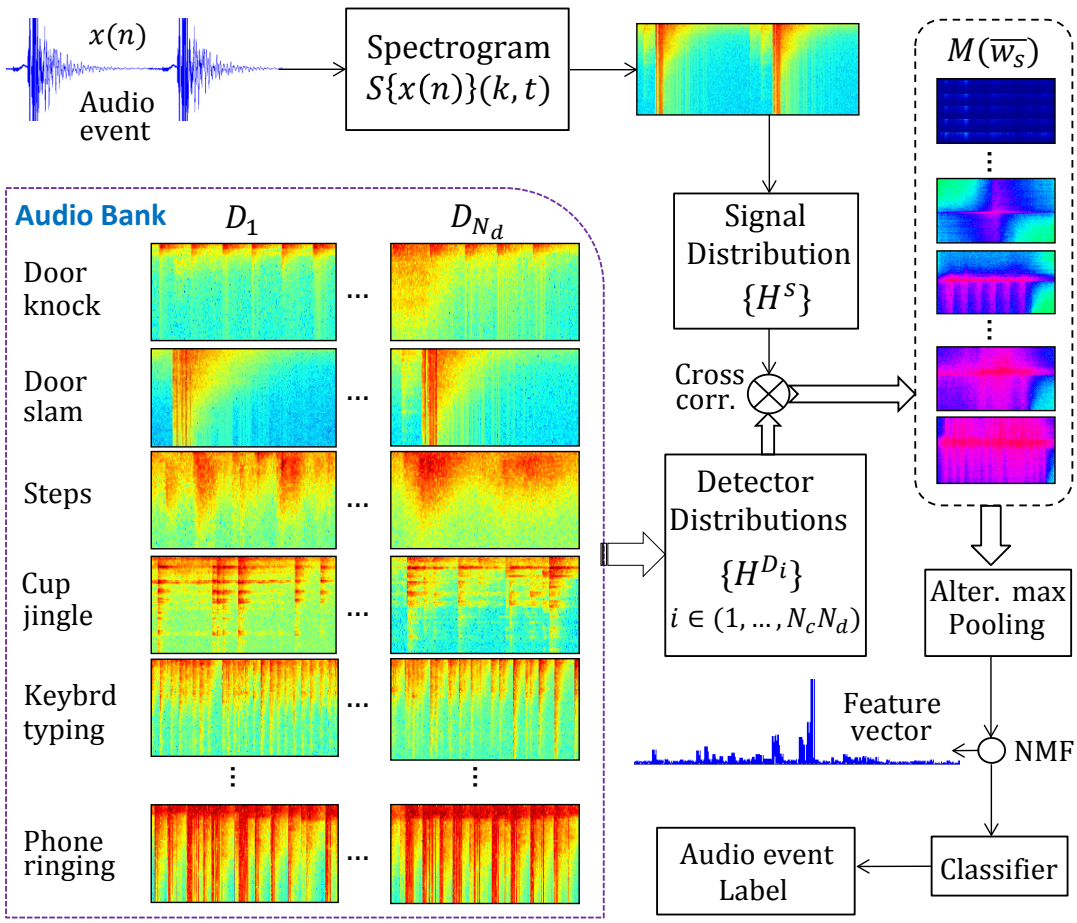


Figure 7.1: Overview of the proposed Audio Event Classification (AEC) framework. Audio bank (sec. 7.2) is the high-level representation of the audio events. It stores a set of audio detectors (sec. 7.2.2) representing each audio class in the spectrogram space (sec. 7.2.1). It produces the feature vector (sec. 7.2.3) by concatenating max-pooled cross-correlation (between detector and signal distribution) patterns from each detector.

Most of the previous works in AER [136], [137], [138], [139] make use of standard features such as mel-frequency cepstral coefficients (MFCC), zero-crossing rate, subband energies, short term energy along with their statistical properties like mean, standard deviation and entropy. But these low and mid-level features are limited in the amount of acoustic semantics they can capture, which yields a representation with inadequate discriminative power, hence none of them is clear winner for robust audio recognition. In this chapter, we propose the Audio Bank, a new high-level representation for audio events, which is semantically rich and highly discriminative in nature. Audio bank explores the set of distinctive audio detectors, which ultimately act like the bases of high dimensional 'audio space' and thereby giving semantically rich representation. Fig. 7.1 gives an overview of the audio bank based feature construction and the AER framework.

The low and mid-level features can not comprehensively represent the underlying harmonic structure in audio data. So conventional audio recognition methods try to uncover the hidden phonic patterns by exploiting data structure using hidden Markov models (HMM) [140] and Gaussian mixture models (GMM) [141]. These models require a lot of training data and are computationally expensive especially for high dimensional feature vectors. Where as the proposed audio bank feature representation is discriminative enough to give good recognition results using even simple classifier like k-nearest neighbors (kNN) [142].

We have applied several learning algorithms in order to measure the AER rate based on audio bank features. Also several experiments are performed by varying the number of bank detectors and the amount of training data to assess the dependency of the proposed framework on various parameters. Our major contributions include; proposing a new high-level and robust feature representation using audio banks and providing an experimental study of its suitability to the classification task by using several learning algorithms. We use UPC-TALP database [8] for audio classification which contains a set of isolated meeting-room acoustic events. Experimental results show that the proposed framework provides better classification performance than the conventional feature extraction methods.

7.2 Audio Bank Representation

Audio signal is composed of intricate phonic patterns which may be repeated several times in the signal. So audio bank tries to exploit this fact and represents an audio as the collected output of many phonic detectors that each produces a correlation pattern. Although, at a glance, audio bank looks related to object bank [143], in detail, the audio classification problem is clearly distinct from the image classification. For the later one, it is evident that image is made up of primary objects like trees, water, mountains, people, buildings etc. So object bank explicitly consists of individual object detectors. But in case of audio signal the atomic phonic patterns from which the audio is formed, are difficult to separate from each other. So we employ a template-based audio detector, where distinctive audio examples themselves serve as a template. To infuse the robustness to loudness (amplitude), we make use of feature distribution similarity measure (Bhattacharya coefficient) for matching the audio detectors. Invariance to pitch (fundamental frequency of an audio) is achieved via analyzing signal in frequency domain. To account for timbre (differences in pitch quality), we sample distinctive templates from the audio space.

7.2.1 Feature Extraction

The spectral content of an audio changes over time, so applying the discrete Fourier transform (DFT) over the entire signal does not reveal transitions in spectral content (non stationary signal). But for short periods of time, audio can be considered to be stationary. So our feature space consists of audio spectrogram, which is collection of the power spectrums of short-time signals as follows,

$$S\{x(n)\}(k,t) = \left| \sum_{n=0}^{N-1} x(n+tM)\, w(n)\, e^{-j\frac{2\pi}{N}nk} \right|^2, \tag{7.1}$$

where $x(n)$ is the audio signal, $w(n)$ is (short-time) analysis window of size N (e.g. Hamming), k is frequency bin index, t is time frame index and M is the framing step (number of samples separating two consecutive frames). Human auditory system is able

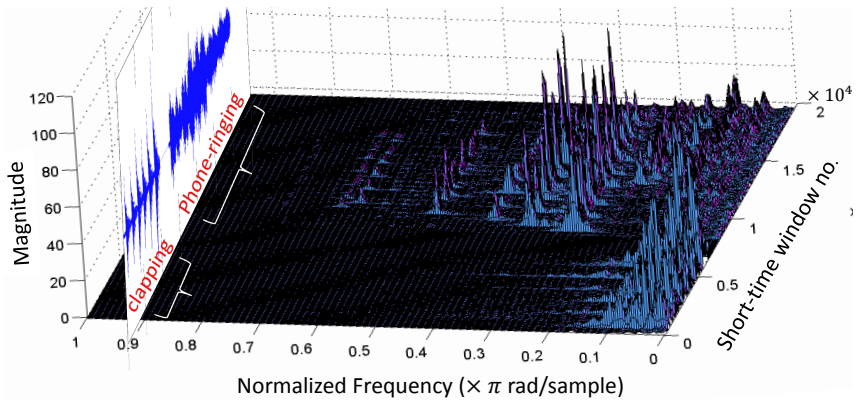


Figure 7.2: Two different audio events (*clapping*, *phone-ringing*) and there respective spectrograms. Both signals show different frequency patterns, where *phone-ringing* class shows much higher frequency content.

to distinguish various audio events because they exhibit typical time-varying frequency patterns. Spectrograms and signal waveforms for two different audio events are shown in Fig. 7.2.

7.2.2 Selecting Bank Detectors

Audio bank gives us great deal of flexibility in choosing the kind of bank detectors; indeed different types of detectors (using different features) can be used concurrently. In our implementation, we use distinctive signal spectrogram as the detector due to its evident capability in localizing events from a single template, efficiency (using FFT [144]), pitch analyzing capability and natural interpretation as an audio decomposition into space-time frequencies. In order to capture discriminative audio patterns, the bank detectors must be distinct from each other.

Let x_i^c be the sample from audio (event) class $c \in \{1, \cdots, N_c\}$. Then for each c, we choose N_d representatives from $\{S\{x_i^c\}|i = 1, 2, \cdots\}$ as detectors $\{D_i\}_1^{N_d}$ by using k-means clustering [145]. To achieve robustness to timbre, we also sample detectors from *audio space* (audios from the same event but having different sound quality), with constraints that the detector size should be small enough to contain only few occurrences

of the underlying audio event. All detectors are stored as spectrograms and thereby producing the bank of size $N_D = N_c \times N_d$.

7.2.3 The Audio Bank Feature Vector

Intuitively the audio bank feature vector is nothing but the concatenation of detection responses of all N_D detectors with the signal feature $S\{x\}$. To obtain the detection response in the large audio feature, the detector is placed at each position and the similarities between the frequency distributions (histograms H^D and H^s) at the corresponding positions of the detector D and the signal feature $S\{x\}$ are computed. Let this similarity be given by $sim\big(H^s(\bar{w}_d), H^D(\bar{w}_d)\big)$, where $\bar{w}_d = (k, t)$ ranges over frequency-time support of the detector. The global similarity match measure, $M(\bar{w}_s)$, at each position $\bar{w}_s = (k, t)$ of the signal feature is obtained by summing the individual similarity measure across the detector as,

$$M(\bar{w}_s) = \sum_{\bar{w}_d} sim\big(H^s(\bar{w}_d), H^D(\bar{w}_d - \bar{w}_s)\big). \tag{7.2}$$

We use the Bhattacharyya coefficient [146] as the similarity measure, because of two reasons; first, it bounds the values between 0 and 1, with 1 indicating complete match; second, it yields to efficient implementation. The Bhattacharyya coefficient for two histograms H^1 and H^2, each with B bins and b as the bin index, is defined as,

$$sim(H^1, H^2) = \sum_{b=1}^{B} \sqrt{H_b^1, H_b^2}. \tag{7.3}$$

Now after inserting measure (7.3) into global match measure (7.2) and rearranging the summation orders produces,

$$M(\bar{w}_s) = \sum_{b} \sum_{\bar{w}_d} \sqrt{H_b^s(\bar{w}_d)} \sqrt{H_b^D(\bar{w}_d - \bar{w}_s)}, \tag{7.4}$$

which is the cross-correlation between the individual bins of the histograms as, $M(\bar{w}_s) = \sum_b \sqrt{H_b^s} \star \sqrt{H_b^D}$. These 2-dimensional correlations are computed efficiently in the fre-

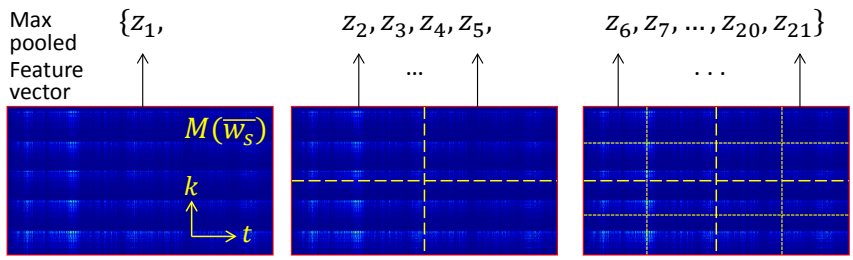


Figure 7.3: Alternate max-pooling operation extracts feature vector from the correlation output of each detector.

quency domain using the convolution theorem of the Fourier transform [147], where the computationally expensive correlation operations are exchanged for relatively simpler pointwise multiplications as follows,

$$M(\bar{w}_s) = F^{-1}\Big\{\sum_b F\{\sqrt{H_b^s}\}F\{\sqrt{\widetilde{H}_b^D}\}\Big\}, \tag{7.5}$$

where $\widetilde{H}_b^D$ is the flipped detector distribution with $F\{\cdot\}$ and $F^{-1}\{\cdot\}$ denoting the Fourier transform and its inverse respectively. The Fourier transforms are realized efficiently by using FFT algorithm. The audio bank feature vector is constructed from the global correlation pattern, $M(\bar{w}_s)$, by using alternate max-pooling operation. In this operation, the output correlated pattern is subsequently divided into $\{2^0, 2^2, 2^4\}$ equal parts and then maximum values from all the parts are concatenated one after another to yield $N_d \times N_c \times 21$ dimensional feature vector for each input audio signal. Alternate max-pooling operation (shown in Fig. 7.3) efficiently and compactly captures the underlying variations in the correlation output. Thereby it tries to find the maximally correlated frequency patterns.

7.2.4 Non-Negative Matrix Factorization (NMF)

NMF is a signal representation method for noise-robust feature extraction in the reduced dimension. It aims to minimize the distance between the original signal and its approximation. Given a non negative $m \times n$ matrix X, NMF decomposes it into the

non-negative $m \times k$ matrix W and the non-negative $k \times n$ matrix H that minimize the following reconstruction error,

$$J(W, H) = \|X - WH\|_F^2, \tag{7.6}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. The column vectors of W are the basis vectors and the values along the column in H denote the contribution of the column vectors of W or in other words, it represents the decomposition of the signal values on the basis matrix W. NMF thus finds the 'parts' based, linear representations of non-negative data as only additive combinations of the bases are allowed. W can be learned according to (7.6) using the algorithm proposed in [148] and then H is learned with a fixed W. In our work X is the matrix of max-pooled feature vectors (see Fig. 7.3). The decomposition of a features on the basis matrix is then treated as the new coded features and is the input to the classifiers.

7.3 Experiments

7.3.1 Dataset Description and Experimental Setting

The UPC-TALP dataset [8] contains 14 classes of audio events that occur in a meeting room environment. The events are recorded such that they do not overlap in the time domain and the recording is done at 44.1kHz sampling frequency. Table 7.1 shows the 14 audio classes present in the database along with their annotating labels. In our work, we do not use the class "unknown" for training and evaluation, because it is not of much significance and contains occurances of silence and noise. The classes "door open" and "door close" perceptually sound similar (because both events produce the same door slam sound), so we mix them to make the unified class "door movement" having total 121 samples. So we are left with 12 classes over which we report our experimental results.

For the feature extraction process, the audio samples were down-sampled by $\frac{1}{4}$ and 256-point discrete Fourier transform (DFT) was taken with a frame overlap of 50 % to compute the spectrograms . For the AER analysis using various classifiers, 60 % samples

Table 7.1: Audio events in the UPC-TALP database with their number of audio samples and total duration (sec).

Event Name	**# Samples \ Time**	**Event Name**	**# Samples \ Time**
Door knock (kn)	50 \ 64	Key jingle (kj)	65 \ 146
Door open (do)	60 \ 66	Keyboard typ. (kt)	66 \ 194
Door close (ds)	61 \ 78	Phone ring (pr)	116 \ 315
Steps (st)	73 \ 250	Applause (ap)	60 \ 212
Chair move (cm)	76 \ 216	Cough (co)	65 \ 85
Cup jingle (cj)	64 \ 187	Laugh (la)	64 \ 118
Paper work (pw)	84 \ 300	Unknown (un)	126 \ 89

from each class were randomly assigned as the training data and the rest 40 % samples were used for testing. No audio from the audio bank is used again for testing stage. Extensive experiments were performed (described in following sections) to validate the robustness of the proposed features.

7.3.2 Audio Bank with Different Classifiers

We applied several classifiers individually on the audio bank features to evaluate their suitability and discriminability for AER. We have used, k-nearest neighbor (kNN) , one-vs-all SVM (SVM-A), one-vs-one SVM [149] (SVM-O), Gaussian process [108] (GP) and Neural Network [150] (NN) for classification. For the kNN, k was set to 5. For SVM-A we used RBF kernel with $C = 150, \sigma = 75$, while $C = 100, \sigma = 60$ for SVM-O. These parameters were determined by 5 fold validation.

GP is completely specified by its mean and covariance function. The mean function for the GP classifier was set as a constant and an isotropic squared exponential kernel was used as the covariance function. The implementation of GP was taken from the GPML toolbox [151], which learns the GP hyper-parameters and predicts the class labels. As GP classifier is essentially a binary classifier, we used one-vs-one approach to extend its functionality for multi-class classification. For Neural Network, the number of hidden layers was arbitrarily set to 300.

For each classifier, we performed the recognition experiment 5 times i.e. for each of the 5 runs the data was randomly assigned into the training and test set. Fig. 7.4(a)

Table 7.2: Simulation time of different classifiers for the audio bank features averaged over the 5 runs, where audio bank size is kept fixed, $N_D = 48$

Classifiers	**k-NN**	**SVM-A**	**SVM-O**	**GP**	**NN**
Simulation Time (sec)	0.49	0.89	46.44	180	10.72

shows the average AER accuracy obtained using audio bank features, the whisker at the top of each bar denotes the standard deviation. We can see that the AER performance is fairly good for a simple classifier like kNN, whereas for more complex classifiers like SVM and NN, the performance is significantly better. In case of GP, however the accuracy suffers possibly due to the high feature dimensionality.

Computational time: Table 7.2 shows the total simulation time taken for testing and training in each of the above mentioned classifiers. As evident, kNN requires the least time due to its simplicity. SVM-O and GP on the other hand are comparatively much slower as these classifiers are implemented in one-vs-one fashion where the computational complexity scales quadratically with the number of classes. The simulation time for SVM-A is quite less (comparable to kNN), while NN classifier is moderately slower. Thus, if we compare all the 5 classifiers; according to the simulation time and the recognition accuracy, then SVM-A is a clear winner as it performs considerably well without much compromise on speed.

Exploring kNN classifier: The dependency of AER performance on the value of parameter k in case of the kNN classifier is assessed by varying k from 1 to 100 in intervals of 10. For each k, the recognition performance was recorded 10 times and the average performance is reported in Fig. 7.4(b) along with the corresponding standard deviation indicated by whisker. From the figure we see that the recognition performance drops with increase in k. Thus, lower the k values, the better the AER accuracy.

Exploring Neural Network:We also analyzed the effect of the number of hidden layers in the NN classifier on the recognition performance. The number of hidden layers was varied from 10 to 1000 and the recognition performance for each hidden layer value was measured by averaging over 10 runs, similar to the kNN experiment. Fig. 7.4(d)

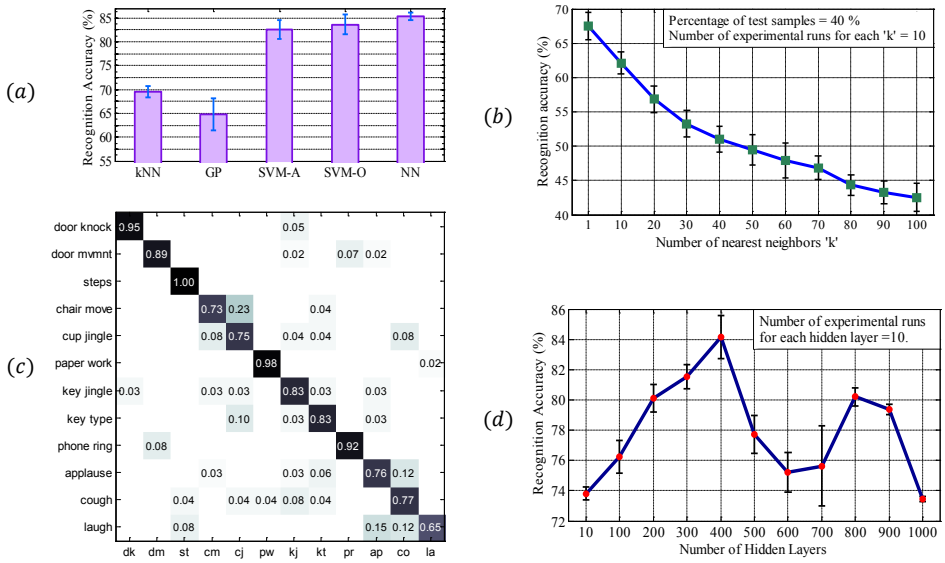


Figure 7.4: (a) AER by Audio Bank using different classifiers, (b) kNN classifier performance with varying k, (c) Confusion matrix (AER rate) using Audio Bank, (d) Neural network performance with varying hidden units.

shows the AER accuracy obtained for each hidden layer value. It shows that if the number of hidden layers is in between 300 and 400, the recognition performance is considerably good and it does not degrade significantly for other number of hidden units.

7.3.3 Comparison with Other Methods

Audio bank (size $N_D = 48$) features are compared with 3 sets of conventional features using the same SVM-O classifier. Feature 'set-A' consists of 32 dimensional (32-D) log-filter bank energies, the zero crossing rate (ZCR) and the signal concatenated into a 34-D feature vector. Feature 'set-B' consists of 2050-D feature vector constructed by using FFT coefficients [147], ZCR and signal energy. And feature 'set-C' consists of 13-D MFCC coefficients [137]. Table 7.3 shows the average AER rate achieved over 5 runs of the experiment. The performance of the audio bank features is significantly better than the conventional features. Its per class performance is portrayed in the form of confusion matrix and shown in Fig. 7.4(c). High diagonal entries signify the interclass

Table 7.3: Conventional feature sets vs Audio Bank feature.

Methods	set-A [136]	set-B [147]	set-C [137]	Audio Bank
AER rate (%)	51.44	44.16	73.38	84.59

discriminability of the proposed features. The method slightly confuses for 'chair moving' class, where it miss-classifies some of the entries to 'cup jingle'. The method achieves perfect recognition rate for 'steps' and more than 90% accuracy for 'door knock', 'paper work' and 'phone ring' classes.

7.3.4 Training Data Size Variation

Amount of training data plays a vital role in deciding the classification performance. More the training data, better will be the generalization performance. So the amount of training data from each class was varied from 10 to 90 % (correspondingly test data per class became 90 to 10 %); then the resulting AER performance is summarized in Fig. 7.5(a). It also shows the instability of NN classifier (due to high dependency on parameter initializations). The performance of the SVM-A and SVM-O gradually increases as the training data increases. We can see that, even 40% of training data per class gives more than 80% AER rate. This shows the discriminability of the proposed features.

The proposed framework is very general and can easily adapt to different AER settings by simply adding more new distinctive detectors to the existing bank. However it is not obvious that a larger bank necessarily gives better performance, because as the number of bank detectors (N_D) increases, dimensionality of the audio bank feature vector also increases and AER rate may hinder due to the curse of dimensionality. To assess these effects we performed AER by gradually varying the bank size. Audio bank consists of distinctive N_d detectors from each audio class (see sec. 7.2.2). We varied the bank size from 3 to 120. The dependency of AER rate (using 60% of training data from each class) with the bank size is shown in Fig.7.5(b). For NN classifier, the slight decay in AER rate with big bank is observed due to the curse of dimensionality and instability of the NN classifier (see sec. 7.3.3). Tiny bank (N_D = 3 or 6) had smaller

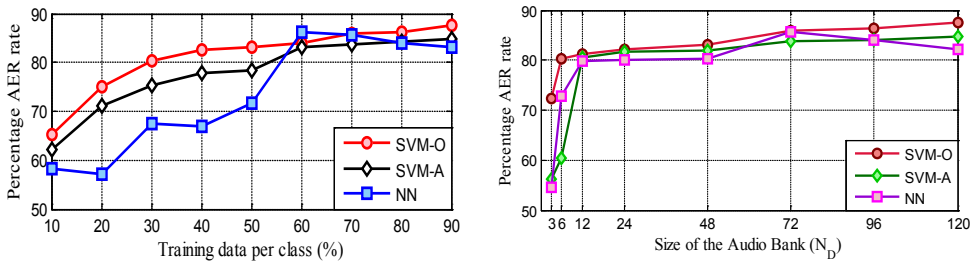


Figure 7.5: (a) Audio event recognition (AER) rate for 3 classifiers with varying training data, (b) Bank size variation and AER rate. Notations: tiny bank (detectors $N_D = 3, 6$), small bank ($N_D = 12$), big bank ($N_D > 50$).

amount of detectors than total classes (12). So some classes were remain unexpressed by the resulting feature vector. Thus tiny bank performed poorly. But even the small bank ($N_D = 12$, one detector for each class) has produced comparable accuracy (80%) to big bank ($N_D > 50$). This surprising stability of the audio bank on its size can be attributed to its ability to represent event in the audio space and thereby producing a high-level representation which is also group sparse in nature. Because for responding to a particular class of audio event, only the detectors that are closely related to it, will produce high detection response; whereas the semantically remote detectors will remain in the dormant state. Thus even after using the small bank, the resulting group sparse and semantically rich feature is powerful enough to produce high AER performance.

7.4 Conclusion

We proposed Audio Bank, a new high-level efficient audio representation for AER, aimed at exploring the underlying harmonic structure present in the audio data. It is comprised of distinctive audio detectors representing each audio class in frequency-temporal space. It produces superior features as compared to low-level features in discriminating audio events, by appropriately pooling a large set of smaller audio detectors. Feature stability on the bank size and high AER performance using several classifiers (SVM, neural network, Gaussian process and k-NN) shows the effectiveness of the proposed method.

Chapter 8

Concluding Remarks

In this thesis, we have proposed the unified framework for Human Activity Recognition (HAR), where the existing methods tackle variety of human activities independently. This complete and unified HAR framework stands on the pillars of newly proposed features and algorithms. The first one (feature) summarizes the very high dimensional video data by encapsulating the underlying motion dynamics succinctly into the low dimensional representative vector. Whereas the other one (algorithm) uncovers the hidden subtleties that lies among the samples of the activity class and helps for clearly inferring true activity class label.

Features play a vital role in HAR. Global features are generated using the entire video sequence while ignoring explicit temporal information but they capture the oriented and holistic underlying patterns. We found that HAR can be improved by fusing extra temporal information with global representation. So we have proposed new mid-level features (Frequencygrams, Spatiograms) by analyzing dynamics of the motion histograms in frequency and spatio-temporal domain; and new high-level features (Abstracted Radon Profiles) by considering whole oriented information of action silhouettes. These oriented holistic features have provided a natural and discriminative representation for reciprocating motions. We also have proposed the *low-level features* (viz., Circulation, Motion Homogeneity, Motion Orientation and Stationarity), to readily capture the pixel level

motion information. Using them, we have achieved the highest performance for gesture (HWU dataset) and abnormal activity recognition (on UMN dataset).

On the algorithmic aspect, we have proposed the Graph Pyramid, a hierarchical graph analysis framework for supervised HAR and the Proximity clustering for fast and unsupervised abnormal HAR. Graph Pyramid has made it possible to uncover the hidden subtleties of the action family by considering interactions among the neighborhood nodes to the query. Machine vision becomes blind in dark illumination conditions and severe occlusions. So we have also incorporated the acoustic information by proposing the Audio Bank, a new high-level semantic representation of an audio, to offer more robustness for various activity recognition.

In case of traffic scenario, we have implemented the time series embedding framework to solve the data scarcity problem for traffic activity recognition. Using multi-task learning framework, we have learnt all activity pattern classifiers simultaneously. We have improved the traffic pattern recognition performance (on QMUL, MIT, NGSIM and our own Wide Interaction traffic dataset) by several magnitude as compared to the state-of-the-art approaches. Overall in HAR domain, the training data scarcity is the common issue. Without dataset rebalancing, the classifier becomes biased toward majority class samples. To address imbalanced dataset problem, we have proposed the G-SMOTE algorithm. It is an improvement to the existing synthetic minority oversampling technique. Its extensive evaluation on several datasets has produced the state-of-the-art results.

The proposed graph pyramid and proximity clustering are the general pattern recognition algorithms and can be applied to problems from various other domains. So it would be interesting to use them for tackling recognition problems like protein sequences classification from bioinformatics, soil type classification from geology etc. It might be necessary to extract different graph topological features for analyzing graph pyramid concerning other domain problems. Hence this thesis opens up many avenues for future research, but for now,

we have taught the machine how to see through the camera eye!

Bibliography

[1] Stauffer C and Grimson WEL, "Learning patterns of activity using realtime tracking," in *IEEE Transactions on pattern analysis and machine intelligence*, 2000.

[2] Haritaoglu I, Harwood D, and Davis LS, "W4: real-time surveillance of people and their activities," in *IEEE Transactions on pattern analysis and machine intelligence*, 2000.

[3] P. Barattini, C. Morand, and NM. Robertson, "A proposed gesture set for the control of industrial collaborative robots," in *RO-MAN*, 2012.

[4] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *ICCV*, 2005.

[5] "Unusual crowd activity UMN dataset,," in *http://mha.cs.umn.edu/Movies/*, 2008.

[6] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," in *ICPR*, 2004.

[7] M.D. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *CVPR*, 2008.

[8] A. Temko, R. Malkin, C. Zieger, D. Macho, C. Nadeu, and M. Omologo, "Clear evaluation of acoustic event detection and classification systems," in *MTPH*, 2007.

[9] T. Hospedales, Shaogang G., and Tao X., "A markov clustering topic model for mining behaviour in video," in *ICCV*, 2009.

[10] H. Wang, MM. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *BMVC*, 2009.

[11] "Ngsim: The Next Generation Simulation and Surveillance dataset," in *Available at http://www.ngsim.fhwa.dot.gov*.

[12] M.M. Rahman and S. Ishikawa, "Robust appearance-based human action recognition," in *ICPR*, 2004.

[13] X. Wu, D. Xu, L. Duan, and J. Luo, "Action recognition using context and appearance distribution features," in *CVPR*, 2011.

[14] K. Marco and D. Joachim, "Temporal self-similarity for appearance-based action recognition in multi-view setups," in *Comp analysis of Images*, 2013, Lec Notes in CS.

[15] J. Yamato, Jun Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden markov model," in *CVPR*, 1992.

[16] C. Sminchisescu, A. Kanaujia, Zhiguo Li, and D. Metaxas, "Conditional models for contextual human motion recognition," in *ICCV*, 2005.

[17] G. Luo and W. Hu, "Learning silhouette dynamics for human action recognition," in *ICIP*, 2013.

[18] T. Sandhan, T. Srivastava, A. Sethi, and Jin Young Choi, "Unsupervised learning approach for abnormal event detection in surveillance video by revealing infrequent patterns," in *IEEE Int conf on Image and Vision Computing New Zealand (IVCNZ)*, 2013.

[19] I. Laptev and T. Lindeberg, "Space-time interest points," in *ICCV*, 2003.

[20] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *IEEE Int Workshop VSPE*, 2005.

[21] H. Wang, A. Klaser, C. Schmid, and C. Liu, "Action recognition by dense trajectories," in *CVPR*, 2011.

[22] JC. Niebles, H. Wang, and L. Fei-fei, "Unsupervised learning of human action categories using spatial-temporal words," in *Proc. BMVC*, 2006.

[23] A. Klser, M. Marszaek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *In BMVC*, 2008.

[24] AF. Bobick and JW. Davis, "The recognition of human movement using temporal templates," *PAMI*, 2001.

[25] Ying Wang, Kaiqi Huang, and Tieniu Tan, "Human activity recognition based on r transform," in *CVPR*, 2007.

[26] Md. AR Ahad, JK. Tan, H. Kim, and S. Ishikawa, "Motion history image: its variants and applications," *Mach Vis App*, 2012.

[27] J. Cai, G. Feng, and X. Tang, "Human action recognition using oriented holistic feature," in *ICIP*, 2013.

[28] T. Sandhan, Hyung Jin Chang, and Jin Young Choi, "Abstracted radon profiles for fingerprint recognition," in *IEEE Int Conf on Image Processing (ICIP)*, 2013.

[29] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

[30] N. et al. Dalal, "Human detection using oriented histograms of flow and appearance," in *ECCV*, 2006.

[31] D. Fleet and Y. Weiss, "Optical flow estimation," in *Math. models in CV*. Springer, 2006.

[32] Wang Y, Huang K, and Tan T, "Human activity recognition based on r transform," in *CVPR*, 2007.

[33] D. Nandy and J. Ben-Arie, "Using the fourier slice theorem for representation of object views and models with application to face recognition," in *ICIP*, 1997.

[34] Akio Tojo Masahiro Kawagoe, "Fingerprint pattern classification," *PR*, 1984.

[35] R.O. Duda and et al., *Pattern Classification*, John Willey & Sons, 2nd edition, 2001.

[36] S. Bomma, P. Favaro, and NM. Robertson, "Sparse representation based action recognition," in *ICIP*, 2013.

[37] A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition," in *CVPR*, 2010.

[38] et al. De Campos, "An evaluation of bags-of-words and spatio-temporal shapes for action recognition," in *WACV*, 2011.

[39] Z Gao, A Liu, H Zhang, G Xu, and Y Xue, "Human action recognition based on sparse representation induced by l1/l2 regulations," in *ICPR*, 2012.

[40] Y Cong, J Yuan, and J Liu, "Sparse reconstruction cost for abnormal event detection," in *CVPR*, 2011.

[41] R Mehran, A Oyama, and M Shah, "Abnormal crowd behavior detection using social force model," in *CVPR*, 2009.

[42] S Wu, B Moore, and M Shah, "Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes," in *CVPR*, 2010.

[43] V Saligrama and Z Chen, "Video anomaly detection based on local statistical aggregates," in *CVPR*, 2012.

[44] Cheriyadat AM and Radke R, "Detecting dominant motions in dense crowds," in *IEEE Journal of Selected Topics in Signal Processing*, 2008.

[45] Brostow GJ and Cipolla R, "Unsupervised bayesian detection of independent motion in crowds," in *CVPR*, 2006.

[46] Efros AA, Berg AC, Mori G, and Malik J, "Recognizing action at a distance," in *ICCV*, 2003.

[47] Haritaoglu I, Harwood D, and Davis LS, "W4: Who? when? where? what? a real time system for detecting and tracking people," in *IEEE Int. Conf. on Face and Gesture Recognition*, 1998.

[48] Wiliem A, Madasu V, Boles W, and Varlagadda P, "Detecting uncommon trajectories," in *Computing: Techniques and Applications*, 2008.

[49] Hu W, Xiao X, Fu Z, Xie D, Tan T, and Maybank S, "A system for learning statistical motion patterns," in *IEEE Tran. PAMI*, 2006.

[50] Jiang F, Wu Y, and Katsaggelos AK, "Abnormal event detection from surveillance video by dynamic hierarchical clustering," in *IEEE Int. Conf. on image processing*, 2007.

[51] Charara N, Jarkass I, Sokhn M, Mugellini E, and Khaled OA, "Adabev: Automatic detection of abnormal behavior in video-surveillance," in *International Conference on Image, Signal and Vision Computing*, 2012.

[52] Davis JW and Sharma V, "Fusion-based background-subtraction using contour saliency," in *CVPR*, 2005.

[53] Andrade EL, Blunsden S, and Fisher RB, "Modelling crowd scenes for event detection," in *International conference on pattern recognition (ICPR)*, 2006.

[54] Mahadevan V, Li W, Bhalodia V, and Vasconcelos N, "Anomaly detection in crowded scenes," in *CVPR*, 2010.

[55] Wu S, Moore B, and Shah M, "Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes," in *CVPR*, 2010.

[56] Niu W, Long J, Han D, and Wang YF, "Human activity detection and recognition for video surveillance," in *ICME*, 2004.

[57] Ng AY, Jordan MI, and Weiss Y, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2001.

[58] Ester M, Kriegel HP, Jorg S, and Xu X, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *AAAI Press*, 1996.

[59] Galluccio L, Michel O, Comon P, Kliger M, and Hero AO, "Clustering with a new distance measure based on a dual-rooted tree," in *Information Sciences*, 2013.

[60] Wei jiang X, Liu sheng H, Yong long L, Yi fei Y, and Wei wei J, "Protocols for privacy-preserving dbscan clustering," in *Int Jr of Security and Its App.*, 2007.

[61] Huang K, Wang L, Tan T, and Maybank S, "A real-time object detecting and tracking system for outdoor night surveillance," in *Pattern Recognition*, 2008.

[62] Zhao Y and Karypis G, "Empirical and theoretical comparisons of selected criterion functions for document clustering," in *Machine Learning*, 2004.

[63] Huang A, "Similarity measures for text document clustering," in *Computer Science Research*, 2008.

[64] "Image categories database by computational vision group at california institute of technology," in *http://www.vision.caltech.edu/Image_Datasets/Caltech6*, 2011.

[65] J. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," in *ICCV*, 2005.

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

[67] W Lin, Y Chen, J Wu, H Wang, B Sheng, and H Li, "A new network-based algorithm for human activity recognition in videos," in *IEEE Tran CSVT*, 2013.

[68] W Li, Z Zhang, and Z Liu, "Expandable data-driven graphical modeling of human actions based on salient postures," in *IEEE Tran CSVT*, 2008.

[69] E. Yu and J.K. Aggarwal, "Human action recognition with extremities as semantic posture representation," in *CVPR*, 2009.

[70] S Yi and V Pavlovic, "Sparse granger causality graphs for human action classification," in *ICPR*, 2012.

[71] AP. Ta, C. Wolf, G. Lavoue, and A. Baskurt, "Recognizing and localizing individual activities through graph matching," in *AVSS*, 2010.

[72] EZ Borzeshi, M Piccardi, and RYD Xu, "A discriminative prototype selection approach for graph embedding in human action recognition," in *ICCV workshop*, 2011.

[73] H Wang, W Wang, J Yang, and PS. Yu, "Clustering by pattern similarity in large data sets," in *In SIGMOD*, 2002.

[74] MCH Devine and BJ. Bohannan, "Phylogenetic clustering and overdispersion in bacterial communities," in *Ecology*, 2006.

[75] A.F. Bobick and J.W. Davis, "The recognition of human movement using temporal templates," in *PAMI*, 2001.

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

[77] X. Sun, M. Chen, and A. Hauptmann, "Action recognition via local descriptors and holistic features," in *CVPR-Wshp*, 2009.

[78] O Pele and M Werman, "The quadratic-chi histogram distance family," in *ECCV*, 2010.

[79] AL Barabasi and ZN Oltvai, "Network biology: understanding the cell's functional organization," in *Nature Genetics*, 2004.

[80] E Ravasz, AL Somera, DA Mongru, ZN Oltvai, and AL Barabasi, "Hierarchical organization of modularity in metabolic networks," in *Science*, 2002.

[81] V Colizza, A Flammini, MA Serrano, and A Vespignani, "Detecting richclub ordering in complex networks," in *Nature Physics*, 2006.

[82] I Gutman, "The energy of a graph," in *Steiermarkisches MathSymposium*, 1978.

[83] Q. Cai, Y. Yin, and H. Man, "Learning spatio-temporal dependencies for action recognition," in *ICIP*, 2013.

[84] Yoo YJ, Sandhan T, Choi JY, and Kim S, "Towards simultaneous clustering and motif-modeling for a large number of protein family," in *IEEE BIBM*, 2013.

[85] Kubat M, Holte R, and Matwin S, "Machine learning for the detection of oil spills in radar images," in *Machine Learning*, 1998.

[86] Provost F and Fawcett T, "Robust classification for imprecise environments," in *Machine Learning*, 2001.

[87] Joshi MV, Kumar V, and Agarwal RC, "Evaluating boosting algorithms to classify rare cases: comparison and improvements," in *IEEE ICDM*, 2001.

[88] Yi L, Hong G, and Feldkamp L, "Robust neural learning from unbalanced data samples," in *IEEE International Joint Conference on Computational Intelligence*, 1998.

[89] Quan Z, Lin gang G, Chong jun W, Wang-jun, and Shi fu C, "Using an improved c4.5 for imbalanced datasets of intrusion," in *Int conf on Privacy, Security and Trust*, 2006.

[90] Sobran NMM, Ahmad A, and Ibrahim Z, "Classification of imbalanced dataset using conventional naïve bayes classifier," in *Int. Conf. on AI in CS and ICT*, 2013.

[91] Kotsiantis S and Pintelas P, "Mixture of expert agents for handling imbalanced data sets," in *Annals of Mathematics, Computing and TeleInformatics*, 2003.

[92] Kotsiantis S, Kanellopoulos D, and Pintelas P, "Handling imbalanced datasets: A review," in *GESTS International Transactions on Computer Science and Engineering*, 2006.

[93] Kubat M and Matwin S, "Addressing the curse of imbalanced training sets: One sided selection," in *Int Conf on Machine Learning*, 1997.

[94] Zheng Z, Wu X, and Srihari R, "Feature selection for text categorization on imbalanced data," in *SIGKDD*, 2004.

[95] Chawla NV, Hall LO, Bowyer KW, and Kegelmeyer WP, "Smote: Synthetic minority oversampling technique," in *Journal of Artificial Intelligence Research*, 2002.

[96] Ma Y, Derksen H, Hong W, and Wright J, "Segmentation of multivariate mixed data via lossy data coding and compression," in *IEEE Trans. Pattern Anal. Mach. Intell.*, 2007.

[97] Pearson K, "On lines and planes of closest fit to systems of points in space," in *Philosoph. Mag., vol. 2, no. 6, pp. 559572*, 1901.

[98] Tamimi H and Zell A, "Global visual localization of mobile robots using kernel principal component analysis," in *IEEE lRSl*, 2004.

[99] Zhao XM, Li X, Chen L, and Aihara K, "Protein classification with imbalanced data prediction report," in *wiley*, 2008.

[100] Landgrebe TCW, Paclick P, and Duin RPW, "Precision-recall operating characteristic (p-roc) curves in imprecise environments," in *ICPR*, 2006.

[101] Daskalaki S, Kopanas I, and Avouris N, "Evaluation of classifiers for an uneven class distribution problem," in *Proceedings of Applied Artificial Intelligence*, 2006.

[102] Huang J and Ling CX, "Using auc and accuracy in evaluating learning algorithms," in *IEEE Transaction on Knowledge and Data Engineering*, 2005.

[103] Andreeva A, Howorth D, Brenner SE, Hubbard TJ, Chothia C, and Murzin AG, "Scop database in 2004: refinements integrate structure and sequence family data (scop40 database, accession number: Pcb00019)," in *Nucleic Acids Res*, 2004.

[104] Bache K and Lichman M, "UCI machine learning datasets," in *http://archive.ics.uci.edu/ml/machine-learning-databases*, 2013.

[105] Saigo H, Vert JP, Ueda N, and Akutsu T, "Protein homology detection using string alignment kernels," in *Bioinformatics*, 2004.

[106] Smith TF and Waterman MS, "Identification of common molecular subsequences," in *J Mol Biol*, 1981.

[107] Needleman SB and Wunsch CD, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," in *J Mol Biol*, 1970.

[108] R. M. Neal, "Regression and classification using gaussian process priors," in *Oxford University Press*, 1998.

[109] R. Emonet, J. Varadarajan, and J. Odobez, "Extracting and locating temporal motifs in video scenes using a hierarchical non parametric bayesian model," in *CVPR*, 2011.

[110] J. Varadarajan, R. Emonet, and J. Odobez, "Bridging the past, present and future: Modeling scene activities from event relationships and global rules," in *CVPR*, 2012.

[111] D. Kuettel, M.D. Breitenstein, L. Van Gool, and V. Ferrari, "What's going on? discovering spatio-temporal dependencies in dynamic scenes," in *CVPR*, 2010.

[112] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," in *PAMI*, 2011.

[113] M.P. Kumar and D. Koller, "Efficiently selecting regions for scene understanding," in *CVPR*, 2010.

[114] A. Basharat, A. Gritai, and M. Shah, "Learning object motion patterns for anomaly detection and improved object detection," in *CVPR*, 2008.

[115] B. Zhao, L. Fei-Fei, and E.P. Xing, "Online detection of unusual events in videos via dynamic sparse coding," in *CVPR*, 2011.

[116] I. Saleemi, L. Hartung, and M. Shah, "Scene understanding by statistical modeling of motion patterns," in *CVPR*, 2010.

[117] V. Mahadevan, Weixin Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *CVPR*, 2010.

[118] X. Li, "Regularized adaptation: theory, algorithms and applications," in *Ph.D. dissertation*, 2007.

[119] S.J. Pan and Q. Yang, "A survey on transfer learning," in *IEEE Trans. on KDE*, 2010.

[120] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell, "Semi-supervised domain adaptation with instance constraints," in *CVPR*, 2013.

[121] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification," in *ACL*, 2007.

[122] R. Caruana, "Multitask learning," in *Ph.D. dissertation, CS, CMU*, 1997.

[123] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," in *Machine learning*, 2008.

[124] J. Xu, G. Ye, and J. Zhang, "Long-term trajectory extraction for moving vehicles," in *IEEE workshop on Multimedia Signal Processing*, 2007.

[125] F. Porikli, "Trajectory pattern detection by hmm parameter space features and eigenvector clustering," in *ICME*, 2004.

[126] X. Wang, K. Tieu, and E. Grimson, "Learning semantic scene models by trajectory analysis," in *ECCV*, 2006.

[127] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," in *PAMI*, 2000.

[128] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent dirichlet allocation," in *Journal of Machine Learning research*, 2003.

[129] F. Taken, "Detecting strange attractors in turbulence," in *Lec notes in Maths*, 1981.

[130] M. Perc, "The dynamics of human gait," in *European journal of physics*, 2005.

[131] A.M. Fraser and H.L. Swinney, "Independent coordinates for strange attractors from mutual information," in *Physics reviews*, 1986.

[132] M.B. Kennel, R. Brown, and H.D.I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," in *Physics reviews*, 1992.

[133] X. Chen, W Pan, JT. Kwok, and JG. Carbonell, "Accelerated gradient method for multi-task sparse learning problem," in *ICDM*, 2009.

[134] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, 2011.

[135] C. Müller, J. I. Biel, E. Kim, and D. Rosario, "Speech- overlapped acoustic event detection for automative applications," in *Interspecch*, 2008.

[136] A. Temko and C. Nadeu, "Acoustic event detection in meeting-room environments," in *Pattern Recognition Letters*, 2009.

[137] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in *European Signal Processing Conference*, 2010.

[138] C. Clavel, T. Ehrette, and G. Richard, "Event detection for an audio-based surveillance system," in *IEEE International Conference on Multimedia and Expo*, 2005.

[139] P. K. Atrey, M. Maddage, and M. S. Kankanhalli, "Audio based event detection for multimedia surveillance," in *IEEE ICASSP*, 2006.

[140] S. R. Eddy, "Hidden markov models," in *Current opinion in structural biology*, 1996.

[141] D. Reynolds, "Gaussian mixture models," in *Encyclopedia of Biometrics*, 2009.

[142] T. Cover and P. Hart, "Nearest neighbor pattern classification," in *IEEE Transactions on Information Theory*, 1967.

[143] L. J. Li, H. Su, E. P. Xing, and F. F. Li, "Object bank: A high-level image representation for scene classification and semantic feature sparsification," in *NIPS*, 2010.

[144] J. S. Walker, "Fast fourier transforms," in *CRC press*, 1996.

[145] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," in *Applied statistics*, 1979.

[146] S.H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," in *Int Jrnl of math models and methods in applied sciences*, 2007.

[147] A.V. Oppenheim and R.W. Schafer, "Discrete-time signal processing," in *Prentice Hall*, 1999.

[148] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," in *Journal of Machine Learning Research*, 2010.

[149] C. Cortes and V. Vapnik, "Support vector machine," in *Machine learning*, 1995.

[150] T. M. Hagan, H. B. Demuth, and M. H. Beale, "Neural network design," in *Boston: Pws Pub.*, 1996.

[151] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (gpml) toolbox," in *Journal of Machine Learning Research*, 2010.

국문 초록

행동 인식(Human Activity Recognition, HAR)은 군중 행동 인식, 사람간 상호작용 분석, 사람의 몸짓과 행동의 인식 등을 포괄하는 컴퓨터 비전과 기계 학습의 다면적인 주제이다. 이 분야는 영상 감시, 보안, 엔터테인먼트, 건강 관리 시스템, 동영상 인덱싱, 인간-컴퓨터 상호작용, 동영상 탐색 등의 넓은 응용 분야에서 수요가 급증하고 있으며, 지난 십 수년간 행동 인식을 연구하는 다양한 방법들이 개발되었다. 본 학위논문에서는 통합된 행동 인식 프레임워크를 만드는 새로운 강인한 특징(feature)과 알고리즘을 제안하여 기존 연구들의 문제점을 해결하고자 한다.

특징은 행동 인식에서 중요한 역할을 한다. 전반적인 특징들(global features)은 전체 동영상의 일시적 정보가 아닌 전체적이고 경향을 지닌 패턴을 수집하여 생성된다. 본 논문은 전반적 표현들(representation)과 추가적인 일시적 정보들을 혼합하여 행동 인식의 성능을 높일 수 있다는 사실에 주목하였다. 본 논문에서는 주파수와 시공간 영역에서 움직임 히스토그램(histogram)을 분석하여 새로운 중간 수준(mid-level) 특징들인 주파수그램(frequencygrams)과 시공간그램(spatiograms)을 제안하고, 행동 윤곽(silhouette)의 패턴을 분석하여 새로운 높은 수준(high-level) 특징인 추상화된 라돈 프로필(abstracted radon profiles)을 제안한다. 이러한 특징들은 카메라 움직임과 작은 가려짐에 강인하고, 반복되는 움직임을 분별할 수 있는 표현(representation)을 제공하며, 또한 본 논문에서 제안된 계층적 그래프 분석 알고리즘인 그래프 피라미드(graph pyramid) 방식을 통해 교사(supervised) 행동 인식에 사용된다. 그래프 피라미드 분류기는 각각의 움직임들을 마디(node)로 표현하여 행동 분류에 대한 그래프를 생성한다. 학습 과정은 수정된 이차-카이(quadratic-Chi) 거리를 통해 그래프 모서리(edge)에 행동 분류 정보를 입력한다. 이 알고리즘은 질의(query)에 대한 주변 마디들간의 상호작용을 고려함으로써 행동 가계(family)의 숨은 세부 사항들을 드러나게 한다.

광학 흐름(optical flow)은 움직임을 묘사하는 기본적인 방식이지만, 가공되지 않은 형태에서는 배경의 잡음(noise)과 카메라의 움직임과 단위(scale) 변화에 민감하다는 단점이 있다. 하지만 가공되지 않은 데이터로부터 구성된 특징들은 행동의 근본적인 움직임을 담고 있으므로 중요하다. 본 논문에서는 픽셀 수준의 움직임 정보를 얻기 위해 광학 흐름을 이용하여 낮은 수준(low-level)의 통계적 움직임 특징들인 순환(circulation), 움직임 동질성(motion homogeneity), 움직임 방향(motion orientation)과 비유동성(stationary)을 제안한다. 이러한 특징들은 제안된 근접 군집화(proximity clustering) 알고리즘을 통해 비교사적(unsupervised) 비정상 행동 인식을 수행하는 데 사용된다. 이 알고리즘의 핵심은 정상 행동들이 비정상 행동들보다 자주 나타난다는 것에 있다. 제안된 특징 공간에서 정상 행동들을 군집화하고 그 외의 것들을 비정상 행동으로 결정한다. 이 알고리즘은 근접 원칙(proximity principle)에 의해 동작하며, 정상 행동 집단의 수를 특정해줄 필요가 없다.

행동 인식 영역에서 몇몇 행동 분류들은 매우 적은 학습 데이터를 가지고 있기 때문에, 데이터 세트의 균형을 재정비하지 않는다면 학습 알고리즘은 다수의 학습 데이터를 갖는 분류들에 편향된 학습 결과를 얻게 된다. 따라서 불균형한 데이터 세트를 다루기 위해 본 논문에서는 소수의 분류에 대해 과다 표집(oversampling)하고 다수의 분류에 대해 과소 표집(undersampling)하여 부츠트래핑(bootstrapping)하는 G-SMOTE 알고리즘을 제안한다. G-SMOTE는 소수 과다 표집 합성 방식에서 기존의 연구에 비해 성능이 향상되었다. 이는 매우 불균형하도록 생성된 데이터 세트에서 수행된 다양한 평가에서 제안된 방법이 가장 높은 인식 결과를 도출한 것을 통해 확인할 수 있었다.

교통 상황에서, 데이터 결핍 문제를 해결하기 위해, 본 논문에서는 처음으로 시계열 임베딩(time series embedding)을 구현하였다. 본 논문에서는 다중 작업 학습 프레임워크에 의해 서로 다른 움직임 패턴간의 상관도를 활용하여 모든 행동 분류기를 학습하였다. 네 개의 공공 장소 데이터 세트에서의 실험 결과는 제안된 방식이 최신 방식들에 비해 교통 패턴 인식 성능이 우월함을 보여준다.

머신 비전(machine vision)은 어두운 조명이나, 가려짐, 혹은 카메라의 시야를 벗어나는 경우에는 성능이 크게 저하된다. 따라서 영상과 함께 음성 정보를 같이 활용하면 행동 인식 시스템의 성능을 향상시킬 수 있다. 본 논문에서는 음성 행동 인식을 위해 음성 정보의 고수준(high-level) 표현으로 음성 뱅크(audio bank)를 제안한다. 제안된 방식은 주파수-시간(frequency-temporal) 공간에서 각 음성 분류를 나타내는 분별력 있는 음성 탐지기로 구성되어 있다. 이 방식은 모든 뱅크 탐지기들의 응답을 하나의 벡터로 축적하여 저수준(low-level) 특징들에 비해 우월한 특징을 생성한다. 뱅크 크기에 따른 특징의 안정성과 높은 인식 성능은 제안된 방식의 효율성을 보여준다.

주요어: 행동 인식 (몸짓; 비정상 이벤트), 특징, 계층적 그래프 분석, 근접 군집화, 불균형한 데이터 세트 처리

학번: 2012 – 23964

이름: 투샤르 산단