8-2016

ACTIVITY ANALYSIS OF SPECTATOR PERFORMER VIDEOS USING MOTION TRAJECTORIES

Anish Timsina
University of Nebraska-Lincoln, timsina.anish@gmail.com

ACTIVITY ANALYSIS OF SPECTATOR PERFORMER VIDEOS USING MOTION TRAJECTORIES

by

Anish Timsina

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professor Ashok Samal

Lincoln, Nebraska

August, 2016

ACTIVITY ANALYSIS OF SPECTATOR PERFORMER VIDEOS USING MOTION TRAJECTORIES

Anish Timsina, M.S.

University of Nebraska, 2016

Advisor: Ashok Samal

Spectator Performer Space (SPS) is a frequently occurring crowd dynamics, composed of one or more central performers, and a peripheral crowd of spectators. Analysis of videos in this space is often complicated due to occlusion and high density of people. Although there are many video analysis approaches, they are targeted for individual actors or low-density crowd and hence are not suitable for SPS videos. In this work, we present two trajectory-based features: Histogram of Trajectories (HoT) and Histogram of Trajectory Clusters (HoTC) to analyze SPS videos. HoT is calculated from the distribution of length and orientation of motion trajectories in a video. For HoTC, we compute the features derived from the motion trajectory clusters in the videos. So, HoTC characterizes different spatial region which may contain different action categories, inside a video. We have extended DBSCAN, a well-known clustering algorithm, to cluster short trajectories, common in SPS videos. The derived features are then used to classify the SPS videos based on their activities. In addition to using NaïveBayes and support vector machines (SVM), we have experimented with ensemble based classifiers and a deep learning approach using the videos directly for training. The efficacy of our algorithms is demonstrated using a dataset consisting of 4000 real life videos each from spectator and performer spaces. The classification accuracies for spectator videos (HoT:

87%; HoTC: 92%) and performer videos (HoT: 91%; HoTC: 90%) show that our approach out-performs the state of the art techniques based on deep learning.

Acknowledgements

I would like to express my gratitude to my advisor, Dr. Ashok Samal, for his guidance, wisdom and support, throughout the discovering, understanding and implementing of this research. I would not be able to complete and enjoy this experience without his help.

In addition, I would like to thank Dr. Jitender S. Deogun and Dr. Massimiliano Pierobon for their time on master’s committee and for their comments and critiques to improve this work.

Also, I would like to thank Bryan Meehan and Todd Duncan of UNL Police Department for providing us with the game surveillance video, which were invaluable for all our experiments.

Lastly, I would like to thank my wife for being one constant through the ups and downs of my graduate school journey.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Understanding and interpreting crowd behavior is a much researched area in a number disciplines including computer vision, sociology, and psychology [1]. Within computer vision research, crowd analysis encompasses a whole spectrum of topics, ranging from crowd counting, anomaly detection, and crowd tracking to classification of an entire crowd based on its overall action [2]. In addition, there is a body of work on crowd analysis, which models and predicts crowd behavior and action based on sociological and psychological factors like ambition and interest, motivation to act and understanding of the immediate environment [3].

Automated crowd analysis is challenging due to factors such as high degree of occlusion, higher object density and complex interaction between the members of the crowd [2]. Additionally, accurate analysis of crowd behavior can also require understanding of crowd psychology [3]. For example, a political rally on the street would be very different from spectators in an arena or pilgrims in a religious ceremony.

Crowds are broadly classified as casual, conventional, expressive and acting [4]. However, events and situations may transform one form of crowd to another. For example, a peaceful protest rally can turn into a violent mob, based on internal, psychological factors like mood and mental state of the crowd or due to socio-political factors such as any feeling of distrust towards or oppression from authority. In most cases, these transformations have significant associated visual cues, such as the pattern of movement and changes in the mood of the crowd. If the visual cues can be extracted using computer vision techniques, the crowd behavior and its changes can be automatically determined.

Automated methods for crowd analysis can provide researchers with critical information about the crowd including its size and density. These measures are useful in several domains including public space design, surveillance, virtual environment design, and other real world simulations. Additional information about the features of the crowd like direction and speed of movement, and emotional states like anger or excitement will be useful in crowd management applications. These complex features can also be used in several domains like social media applications, smart hardware and software, and security and disaster management. Law enforcement agencies can use anomaly detection techniques to discover and prevent unlawful and harmful activities in a crowd.

1.2 Spectator-Performer Space (SPS)

In this research, we focus on a class of videos that are characterized by a large number of people viewing and/or interacting with a relatively small number of people in a spatially structured environment. We define *Spectator-Performer Space (SPS)* as a crowd dynamics composed of one or more central performers, and a peripheral spectator crowd. Examples of such crowd interactions include (a) entertainment space where performers such as singers, musicians, and actors perform on a stage, (b) sporting events, where the sportsmen and women play in a confined space and (c) civic discourses, where a single speaker or a set of speakers occupy a distinct space (stage or podium) from the crowd. In general, performers occupy a central position within the SPS space. In contrast, the spectators while an integral part of SPS play a secondary role and behave in response to the actions of the performers.

Spectator-Performer Space is composed of two different types of spaces: *Spectator Space (SS)* and *Performer Space (PS)* that have fundamentally different characteristics as summarized below. These form the basis for their understanding including classification.

- **Location:** Performers not only play a central role but also occupy a central and prominent location within the SPS. The space for the performers is clearly delineated and is kept separate from the spectators. While there are many different configurations of SPS, two most arrangements are: (a) concentric: the performer space is surrounded by the spectator space (e.g. sporting events) and (b) opposite: the performers and spectators are facing each other (e.g. political events). The space occupied by the performers and spectators are called the performer space (PS) and spectator space (SS), respectively.
- **Density:** The spectator and performer spaces are also different in their density. Usually the performer space is significantly smaller than that of the spectator space. The number of spectators on the other hand is generally several orders of magnitude

bigger. Thus, the spectator space is more congested and occluded in comparison to the performer space.

- **Motion:** The motions of the performers are characterized by alternating periods of *activity* and *inactivity*. We define period of activity as the time when the performers are engaged in what the spectators have gathered to view as the *primary performance.* We define period of inactivity as the time in between the periods of primary performance. The motion patterns of the performers are generally confined to the PS and are more dynamic. In contrast, the motion patterns of spectators are slower and more diverse in their spatial scope ranging from small movements confined the space occupied by the spectator to the large movements in the spectator space. A list of primary performance for some of the SPSs is summarized in Table 1.1.

Table 1.1: Examples of SPS and their characteristics

Event Type	**Performers**	**Spectator-Performer Space (SPS)**	**Primary Performance**	**Performer Movements**	**Spectator Movements**
Performing Arts	Singers and Musicians, Dancers	Stadiums or Covered Halls.	Singing by the Artist/Band	Complex dance moves, general singing movements	Cheering, dancing, singing and imitating the performers
Sporting Events	Players	Sports Stadium including the stands, sidelines and the field.	Game period when game-clock is running down the game is being contested.	Different form of plays depending on the sports like kicking, dribbling, passing etc.	Cheering, Waves, Jeering
Civil/Political Discourses	Political and Social leaders	Arena, Stadiums, Open field, Picket lines, Moving rallies.	Speeches, Display of civil disobedience	Talking, Waving, Walking, Running	Cheering, Walking

While there is now a significant body of literature in automated methods for analysis of crowd images/videos, research on this class of videos is scarce. For example, Zhan et al [1], provide a comprehensive survey of crowd analysis work, but do not find *any* work that makes this distinction. The majority of work in crowd analysis either focuses on a single type of crowd [2,5] or try to be as generic with the type of crowd as possible [6]. In this thesis, we will focus on spectator-performer videos and present algorithms to delineate the spectator and performer space as well to classify the activities of the actors in both the performer space and spectator space.

1.3 Motivation

The main goal of our research is to develop techniques to analyze SPS activities. Since most of the work in crowd analysis focuses on generic crowds only [2,5,6], they are not immediately applicable and efficient for videos in this space. Ultimately, we want to come up with effective techniques to classify the various types of activities in SPS better. Since many video classification and analysis techniques are computationally expensive, we want to develop techniques which are efficient in both time and space. This research has many diverse applications including the following:

- *Surveillance*: Classification of activity performed by spectators can be useful for surveillance and security. For example, the outlier behavior of individuals in a crowd may be suspicious and may need closer monitoring.
- *Crowd Management*: Spectators have emotional response to the activities of performers expressed with actions like cheering, jeering, clapping and singing. Identifying the *mood* of the crowd, will help in the management of the crowd. There

are many examples of peaceful crowd turning violent [7]. Determination of the emotion of the crowd and the level of its excitement and its changes (e.g. from passive to angry) can assist in crowd management.

- *Performance Analysis*: In some domains, identifying the periods of activity (and inactivity) of the performer(s) can be very helpful in the analysis and subsequent improvements and refinements of the performance. This is particularly applicable in domains where there are many periods of activity in a performance (e.g. sports). Identifying the episodes of inactivity can assist in planning for broadcasting of the events and presentation of advertisements.

1.4 Problem Definition

This thesis describes the approaches to classify videos in the Spectator-Performer Space. Specifically, we define the following two sub-problems and give a formal problem definition in Section 3.1.

(a) Given a fixed camera video of a crowd watching a football game, classify it as *Active* or *Passive*. Active refers to the crowd that is actively cheering, booing, clapping and actively reacting to the football game. Passive refers to the crowd that is not cheering, booing or clapping.

(b) Given a fixed camera video of a football field, classify it as *Play* or *No-Play*. Play refers to situation when the football play like running, passing, and tackle is being made. No-Play refers to the dead-ball situation when the players are not making a football play, and are involved in other activities like time-out, players change, discussion and rest.

1.5 Overview of the Approach

Our classification approaches as based on motion trajectories. The actions of the performers and spectators are represented as motion trajectories. They form the basis for our algorithms that leverage their spatial and temporal properties for classification. The classification techniques are as summarized as follows:

(a) *Motion trajectories*: The activities of the performers and spectators serve as the basis of different classification tasks. Specifically, the first order statistical features from the trajectories are used. We specifically examine the length and orientation of trajectories and show that they can effectively characterize different types of actions.

(b) *Motion trajectories clusters*: Actions of performers induce similar reactions from individuals from crowd resulting in similar movements. Therefore, we find and leverage the cluster trajectories [8] for classification of videos as well.

(c) *Bag-of-words*: We have developed a bag-of-words [9] approach to build the feature vector, that is used in conjunction with a number of individual and ensemble based classifiers [10].

(d) *Deep learning*: We have compared the efficacy of our algorithms with deep learning based classifiers, which uses three-dimensional convolution in the deep learning architecture. We show that the efficacy of our algorithms are better than deep learning based classifier for our dataset, as well as demonstrate the time and resource efficiency of our approach over deep learning.

1.6 Contributions

As mentioned before, there is a scarcity of work in the video analysis space that deal with performances. Our work attempts to address this. Specific contributions of our work include:

(a) We define a new class of crowd, Spectator-Performer Space (SPS) characterized by the spatial dichotomy between the performer(s) and the spectators. We have summarized its properties as well as the fundamental problems in this space.

(b) We have developed novel algorithms to classify video segments into different categories based on the activities of the crowd. The algorithms are based on novel trajectory based features based on motion trajectories. The efficacy of the algorithms has been demonstrated with a large collection of videos from the sports domain.

(c) We have compared our trajectory based classification technique with deep learning classification using three-dimensional convolutional networks.

1.7 Thesis Outline

Rest of the thesis is organized as follows. In Chapter 2, we survey the previous work in the field of crowd analysis, and analyze their strengths and shortcomings. We also examine several deep learning approaches used for video classification. In Chapter 3, we describe the methodology used in our research. Details of feature extraction and classification methods that we have used are presented. In Chapter 4, we present experimental results along with a discussion of the efficacy of the algorithms. We also compare our results with some of the state of the art systems. Finally, in Chapter 5, we conclude with a summary and recommendations for future research in this domain.

Chapter 2

Literature Review

As mentioned before, there is very little research focused on spectator-performer videos. In this chapter, we review the related work more broadly in the domain of crowd analysis and video classification. In Section 2.1, we review body of work on crowd analysis from the perspectives of computer vision, sociology and psychology. In Section 2.2, we summarize the research in video classification with focus on trajectory features and clustering and deep learning.

2.1 Crowd Analysis

Crowd analysis research intersects several broad fields including computer vision, sociology and psychology. The body of work on crowd analysis within computer vision includes detection of crowds, modeling of crowd behavior [2,11] and motion pattern analysis [6,12], anomaly and action detection [11,13], object and pedestrian detection [14] and tracking [6]. Our work focuses on spectator-performer crowds characterized by a sparse crowd of performers (e.g. sports team in a field or stage performer) and a dense crowd of spectators (e.g. people watching from a stadium or a hall). Since the crowd specific analysis is novel, we analyze other types of crowd in order to identify approaches that might be useful for our work [5,6,14]. In the next three subsections, we examine body of work in three major areas of crowd analysis research: object recognition, density measurement and counting, and tracking.

2.1.1 Object Recognition

Object recognition refers to the automatic detection of different types of objects in a crowd video or an image. Early work on object detection focused on detection of humans in a crowd derived from detection of body parts (face, head, etc.) and generalized into detecting pedestrians and the full human body. Most of the detection algorithm used some form of supervised learning, by training on features like histogram of gradient [14], or motion boundary histogram [15].

Since crowded scenes invariably have partial occlusion, the complexity associated with detecting people becomes more difficult and requires more sophistication. Wang et al [16] proposed a mixed HOG-LBP (Histogram of Gradient - Local Binary Pattern) approach to handle partial occlusion. They combine a global object detector that scans entire frame (or image) for humans, with a localized object detector. The localized object detector assigns probability that (a) local area is occluded, and (b) the occlusion hides a human. Several researchers have proposed the use of the bag-of-words [17] which is an order independent feature descriptor approach to detect objects in a crowded scene [15,16]. If sufficient images are available to derive a comprehensive list of all features, i.e. the vocabulary, it is possible to efficiently and accurately represent and identify objects with the vocabulary.

2.1.2 Density Measurement and Counting

Crowd density measurement and counting is critical in modern surveillance systems. Many crowd related mishaps in history have occurred in sporting events, religious gatherings and mass demonstrations [18] and the ability to automatically estimate the count and density of the crowd would assist in its management.

Some of the earliest work in crowd density measurement was based on simple counting. This included counting the number of human faces/body parts in a scene and averaging the over the scene to estimate the crowd density. These methods were less accurate due to heavy occlusion in scenes with crowds, which makes counting human/people difficult. Polus et al [13] categorize crowd based on the density of a crowd into (a) free (b) restricted (c) dense (d) very dense and (e) jammed. This categorization and approaches based on this classification scheme look at the crowd density problem from global view, i.e. density of the overall crowd only. The approaches do not consider the variability of density in the crowd itself; a crowd can be denser in one region and sparse/less dense in another.

Fradi and Dugelay [5] propose estimating crowd density by measuring pixel level data instead of analyzing the whole image (frame) as a whole. Their approach develops crowd density maps as probabilistic density functions enabling the calculation of crowd density in different regions in the crowd. This density map can then be used in conjunction with other surveillance techniques for better crowd understanding.

2.1.3 Crowd Tracking

In general, crowd tracking is used to develop efficient techniques to track individual in crowds accurately [6,12]. Ali and Shah [6] propose a framework based on scene structure to track and predict pedestrian position. They show that in a structured high-density scene, a pedestrian's motion can be defined as a function of global and local forces in that particular scene, i.e. motion of the entire crowd with respect to an external reference point and motion of people inside a crowd with respect to one another.

Rodriguez et al [12] extend the work done by Ali and Shah [6] to include unstructured crowd. They propose multiple models to describe crowd scenes with multiple dominant motions and leverage global information about a crowd like density and structure to determine an energy optimization function. This function is combination of a crowd density estimate and the likelihood of finding individuals in different locations in a crowd. The optimization process maximizes the probability of finding people in a location while minimizing the density estimate of that location. This approach is quite useful in crowd tracking even with the heavy occlusion, common in high-density crowds.

Analyzing the behavior of an entire crowd is a different problem from that of tracking a single person in a crowd. Saxena et al [11] propose a crowd modeling technique based on the type of crowd being analyzed and suggest using different variants of crowd models based on the scenario the crowd depicts. The crowds can be mobs with seemingly random behavior, organized and slow rallies or dense hordes of people in public spaces like bus or train station. They use KLT tracker [19] to determine significant feature points in any frame and then compute crowd motion vectors from those features points. Crowd mobility, speed and direction, calculated and processed from KLT tracker, are then used to develop and train tracking models.

2.2 Video Classification

Video classification refers to automatic classification of action being performed in a video. Most video classification research focuses on single actors in videos, e.g. sports played in the video games [15,20], or specific acts in movies [14]. There is a limited body of work which focuses on classification of the activity of a crowd, or of all the people inside the video [6,13]. In the sections below, we summarize several approaches in classification of video, for both single actors and crowd videos.

2.2.1 Trajectory Based Approach

Trajectory based approaches are very common in video classification as most video classification problems deal with some form on motion in the video. Trajectories can represent the motion of different objects in a video. In many cases, the motion patterns in different categories of video are very different, and can be represented by the trajectories of the objects in motion.

Wang et al [15] propose a set of complex features based on dense sampling of trajectories and motion boundary histogram to classify actions performed in a video. They train support vector machines (SVMs) with the dense trajectories and motion boundary histogram, using a bag-of-words approach [9). This effectively encodes both spatial and temporal information in the histogram of words (features) in a particular video. Using dense trajectories and bag-of-words approach, the researchers report up to 89% accuracy in action detection in videos. In particular, the classification with dense trajectory based descriptor perform with 89.8%, 67.5%, 75.4% 87.8% accuracy in KTH, YouTube, UCF sports and IXMAS data sets respectively. These results are also better than KLT Trajectories, SIFT Trajectories and Dense Cuboids [8].

2.2.2 Deep Learning

Deep learning refers to the approach of using multi-layered non-linear architecture in context of machine learning. Deep architectures are used in order to effectively model the structural and semantic concepts describing complex objects like image, video and audio [21]. In traditional classification approaches, a set of hand-made and predefined features is used to describe an object and the features are subsequently used to train a classifier. In deep learning, however, a hierarchy of feature extractors is used in several layers to create a *pixel-to-classifier* architecture. The features are automatically determined and are neither predefined nor handpicked.

Deep learning has been effectively used in many domains including for image understanding and classification with high accuracy [22]. These classification methods out-perform other hand crafted feature based classifiers in many different applications evaluated with many datasets [2]. The drawbacks of deep-learning classifiers are (a) higher time and space requirements and (b) lack of availability of large volume of labeled training data. In case of image classification, both of these issues are fast disappearing with the availability of huge data sets of images that are already classified in the Internet using games and captchas [23]. Video classification is the new frontier in the applications of deep learning with many researchers and practitioners in industry implementing and testing different deep learning models.

Deep learning for image and video recognition uses c*onvolutional neural networks* (CNN), which are inspired from the visual cortex of cat. A cat's visual cortex contains complex arrangement of cells, and is divided into several smaller regions, where each region of the cells processes only a specific portion of the image [23]. Lecun et al [24] introduced the concept of CNNs and used them for handwriting recognition.

There has been a progression of work, from using the sparsely connected, image-to-classifier architecture in image classification [23,25] to extending these architectures for video classification [20]. Karpathy et al [20] use a multiresolution, foveated architecture to extend the concept of CNN from image classification to video classification. Tran et al [26] extend 2-dimensional CNN to 3-dimensional CNN for video classification. They have used a deep 3-dimensional CNN (C3D) for spatio-temporal feature learning and test the network on UCF1-101 dataset [27]. The 3D CNN better preserves the temporal information of the videos because of 3D convolution and 3D pooling, thereby improving accuracy over 2D convolution.

2.2.3 Trajectory Clustering

In computer vision research, trajectory clustering is a common form of analysis in surveillance and anomaly detection [28], tracking [29], pedestrian counting [30] and motion prediction [31]. There are several approaches to trajectory clustering including density based clustering [32] and partition and group framework [8]. Density based clustering is useful to find clusters of spatially close motion trajectories, which is important in tracking and route prediction. This is important in the analysis of SPS videos, particularly in spectator space as it generally covers a large area, and analyzing trajectories based on spatial distribution may reveal important information about spectators in different regions of the video. Partition and group framework enables long and complex trajectories to be divided into simpler sub-trajectories. As it is highly unlikely that entire trajectories are similar to each other, this helps in finding and grouping overlapping sub-trajectories in case entire trajectory are not similar [7].

Liu et al [32] introduced Tra-DBScan, which partitions trajectories into smaller sub-trajectories and uses DBSCAN, a well-known density based clustering algorithm, with a custom distance measure to form clusters. Lee et al [8] partition longer trajectories into smaller sub-trajectories using the minimum description length principle (represent the trajectory with the best compression) and group the sub-trajectories into clusters using a density based clustering algorithm similar to DBSCAN. They finally generate representative trajectories for each of the clusters by using a sweep line approach averaging the coordinates of points in each line in the trajectory that intersects with equidistant vertical lines.

Our approach to cluster trajectories is similar to both Liu et al and Lee et al, but we use parallel and perpendicular distance only, and separate thresholds to determine how close the trajectories are to each other. Additionally, we use the clusters to find out the actions being represented by the different clusters.

2.2.4 Other Methods

Laptev and Lindeberg [33] introduce the concept of *Space-Time Interest Points* (STIP), which extends the Harris corner detector [34] to incorporate time. Instead of just measuring high variation in space only, the authors recognize the area in the space-time continuum that has highest variation, i.e. finding spatial-temporal corners. There is a body of work, which uses both STIP/ space-time descriptors, and bag-of-words approach to perform action detection and classification in videos [35-37].

Hanna et al [38] propose a Hidden Markov Model (HMM) based video classification approach, which utilizes color-based features. These features are built by calculating the *speed* of change in color from one frame to next. The training is done by employing Baum-Welch algorithm [39] for parameter estimation of HMM. Unknown samples are classifying by first computing the color-based feature and feeding them to the model which then calculates the log-likelihoods of the classes.

Chapter 3

Methodology

In this chapter, we will present our approach to classify the SPS videos. The two proposed trajectory-based features and the algorithms to compute them are described in detail. Various preprocessing steps as well as an analysis of the algorithms are also presented.

3.1 Problem Definition

The problem addressed in this research is to classify SPS videos into specific categories of actions. Formally, the problem can be defined as follows: Given a set of *n* distinct videos $V= \{V_1,V_2,V_3 \ldots V_n\}$ that are either all in spectator space or all in performer space, and set of *m* classes $C =\{ C_1, C_2,C_3 \ldots C_m\}$, train a classification function $F: V \rightarrow C$ to predict class for a new video into one of the *m* classes.

3.2 Overall Approach

We have developed two different trajectory based approaches for video classification in SPS. These approaches are based on (a) first order properties of the individual trajectories and (b) properties of trajectory clusters. The overview of our methodology is presented in Figure 3.1.

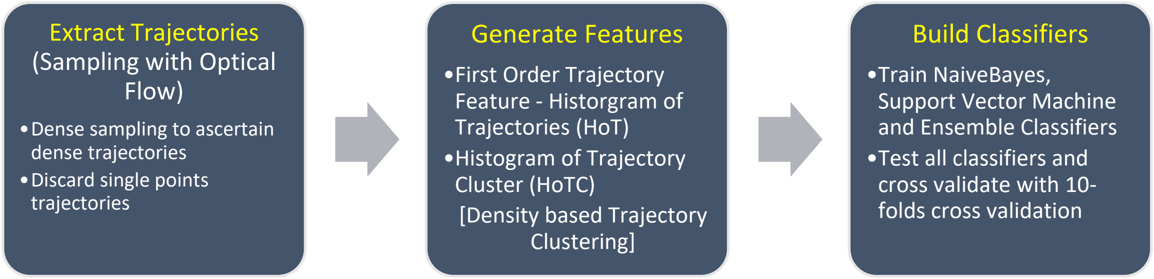
Extract Trajectories
(Sampling with Optical Flow)
•Dense sampling to ascertain dense trajectories
•Discard single points trajectories
Generate Features
•First Order Trajectory Feature - Historgram of Trajectories (HoT)
•Histogram of Trajectory Cluster (HoTC)
[Density based Trajectory Clustering]
Build Classifiers
•Train NaiveBayes, Support Vector Machine and Ensemble Classifiers
•Test all classifiers and cross validate with 10-folds cross validation

Figure 3.1: High-level description of our video classification process

We use optical flow based dense trajectory extraction (described in Section 3.3) to obtain the motion trajectories in a video. This is a widely used trajectory extraction approach [15]. The extracted trajectories form the basis of two novel approaches to generate histogram-based features for classification.

- *Histogram of Trajectories* (HoT): This feature is based on first-order statistics of the motion trajectories, which are computed for each video. Sample features include the length and orientation of the motion trajectories, for the entire video.
- *Histogram of Trajectory Clusters* (HoTC): The trajectories are first grouped to determine spatio-temporal clusters and then the properties of the clusters are used as features. We have extended DBSCAN to determine the clusters. Sample features include the length and orientation of the motion trajectories of the spatio-temporal clusters.

After the features are computed, we train a classifier to map a video to specific activity classes. Previous trajectory based approaches use trajectory vectors generated from optical flow to train bag-of-words based classifiers after generating a dictionary of visual-words based on sample videos. [15]. We take a similar approach to build a visual dictionary of motion trajectories features like length and orientation and use a bag-of-words approach to build classifiers for both spectator and performer videos.

3.3 Trajectory Extraction

We used the approach proposed by Wang et al [15] to extract the dense trajectories from the videos. Feature points are sampled in the first frame of a video, based on a sampling density value (*W*) that represents the number of pixel per sampling point. Sampling more densely, i.e. smaller *W*, results in dense trajectories and vice-versa. Since SPS videos have significant occlusion, we set capture trajectory very densely.

Then, feature points are tracked over successive frames using the optical field approach. These points were tracked for the maximum of 15 consecutive frames. Since objects in the scene move at different speed and for different durations, the trajectories also have variable lengths. Repeating this process for each sample window (feature detection and tracking) generates the trajectories in the video. The result of trajectory extraction phase is a set of trajectories representing the motion induced in the videos. The schematic diagram for trajectory extraction is given in Figure 3.2 and the algorithm in Figure 3.3.

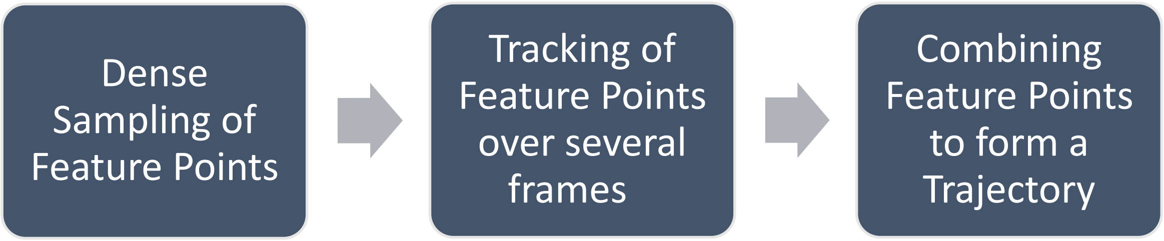
Dense Sampling of Feature Points
Tracking of Feature Points over several frames
Combining Feature Points to form a Trajectory

Figure 3.2: Illustration of the process to generate dense trajectories

```
Function Trajectories_Extraction ($V$, $W$)

Input:   $V = \{F_1, F_2, \ldots, F_n\}$ Video with $n$ number of frames
         $W$ = Number of pixel in a sampling window
Output: $T$= Set of all motion trajectories in the video $V$.

Begin
    Initialize $T = \phi$ // Empty Trajectory Set
    Initialize $P = \phi$ // Empty Point Set

    while ($V \neq \phi$) //There is frame $F_i$ in $V$
        $V = V - \{F_i\}$ //Remove the next frame $F_i$
        $P_{new}$=Sample_New_Points(Fi) //Sample points in current frame
        $P=P_{new} \cup P$ //Add $P_{new}$ to the set of all sampled points

        for each $P_k$ in $P$
            $t_k$=Create_NewTrajectory($P_k$) /* Creates trajectory with starting
                point $P_k$ if it doesn't exist, else returns pre-existing trajectory */
            $T=t_j \cup T$ //Add to return trajectory set
        end for

        for each $P_j \in P$
            $\omega_j$=Optical_Flow_Field($F_i$, $F_{i+1}$)) //Calculate optical flow field
            $t_j$= Track_Points($\omega_j$, $P_j$) //Track point and add to trajectory
        end for

    end while
    return T
end
```

Figure 3.3:Algorithm to generate dense trajectories using dense sampling

Complexity Analysis: Since we sample and track feature points from the start of the video (first frame) to the end (last frame) in the video, the complexity of this algorithm depends on video properties like number of frames and resolution of the video, and sampling window. Let us assume that the video dimensions are $m \times n \times p$, where the spatial resolution is $m \times n$ and there are p frames. Also, let the size of the sampling window be W, usually a small constant.

The maximum number of feature points that can be tracked is based on the sampling window and is given by $\#P=\frac{\#pixels}{W}=\frac{mn}{W}\approx mn$. Optical flow computation is $O(mn)$ for 2D images [40]. The optical flow computation and feature detection and tracking is performed for all frames in the video (Line 12. In Figure 3.3). Thus, the overall complexity of the algorithm is $O(p \times mn)=O(mnp)$.

The number of feature points in the video binds the space requirements. Since number of feature points in a frame is $O(mn)$ (as explained earlier), the space complexity is $O(p \times mn)=O(mnp)$.

3.4 Histogram of Trajectories (HoT)

The first set of features for our classification is based on histogram properties of individual trajectories. Specifically, we examine two key properties of the trajectories – orientation and length. In this section, we describe them in detail present algorithms to compute them.

3.4.1 Orientation

In both spectator and performer videos, motion trajectories vary based on the kind of action being performed. A player running horizontally through a filed would form several horizontal trajectories whereas a spectator making a wave or cheering would form a more slanted trajectory. We define trajectory orientation based on the angle at which the motion trajectory is, with respect to horizontal axis.

Trajectory direction is the angle made by the trajectory with the horizontal axis (x-axis). For the purpose of our analysis, since we were interested in how vertical or horizontal the trajectories were, we evaluated the direction to be in the range of *0-180*°.

For a trajectory t, we calculate the angle made by it with the x-axis as

$$\theta = \arctan\left(\frac{t.endpoint.y - t.startpoitn.y}{t.endpoint.x - t.startpoint.x}\right) \text{----------------------------} \textit{Equation 3.1}$$

Then, we define the direction of the trajectory as

$$Direction(t) = \theta \text{ , if } 0 < \theta \leq 180, \quad \text{---------------------------} \textit{ Equation 3.2}$$

$$\theta\text{-}180, \textit{ otherwise.}$$

Since, direction is a continuous variable, we discretize it by further processing. We define orientation-bins *(Δ_θ)*, which are the range of orientation with equal width, and total number of orientation-bins (*g*), which is a positive integer greater than or equal to 1 with the relation, $\Delta_\theta \times g = 180°$. Finally, we define a function *Orientation* to calculate orientation of a trajectory t given by

$$Orientation(t) = Ceiling\left(\frac{Direction(t)}{\Delta_\theta}\right) \text{-------------------------} \textit{Equation 3.3}$$

Hence, any motion trajectory will have one out of g different orientations. The illustration of this calculation process *g=9* (*or* Δ_θ*=20°)* is given in Figure 3.4.

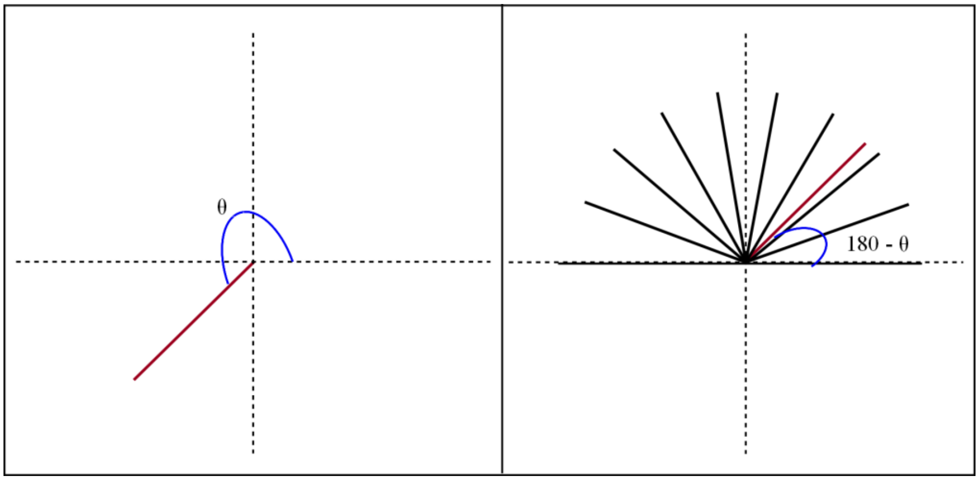

Figure 3.4: Illustration of the process to calculate orientation of a trajectory

3.4.2 Length

Trajectory length varies across different classes of video due to different factors like density, occlusion, and the amount of motion in a scene. Some videos contain scene with high of overall motion but only short motion trajectories due to trajectory fragmentation, whereas other videos can have longer motion trajectories due to little or no occlusion. Length of a trajectory is therefore an important characteristic of video scenes.

We define absolute-length $len(t)$ of a trajectory using the formula given below.

$$len(t)=\sqrt{(t.endpoint.x - t.startpoint.x)^2+(t.endpoint.y - t.startpoint.y)^2} \text{ --- Equation 3.4}$$

This is the Euclidean distance between the start and the end-points of the trajectory. We also define *Maxlen* as the maximum possible value of $len(t_i)\ \forall t_i \in T$, where T is the set of all trajectories in a video. Since absolute-lengths are continuous, we define a discrete measure and name it *Length-category*. We define Δ_{len} as the length-bin, h as the total number of length-category, with relation $\Delta_{len} \times h = Maxlen$. Finally, we define a function to calculate *Length-category* of a trajectory t as

$$Length(t)=Ceiling\left(\frac{len(t)}{\Delta_{len}}\right)\text{--Equation 3.5}$$

Consequently, a motion trajectory can have one out of h different length-category.

3.4.4 Histogram of Trajectory (HoT)

There are several histograms based features which are used in image and video based classification. Histogram of Oriented Gradients (HOG) [14], Histogram of Optical Flow (HOF) [41], and Motion Boundary Histogram (MBH) [41] are all features used for video classification [15]. These features are effective in representing different activities in a video with individual actors, or videos of lower density. Each of these features are effective in visually representing the motion and shapes in videos and images[15]. Since, SPS videos are have higher density, these features won't be effective. However, motion trajectory length and orientation vary significantly with different types of activities in videos. So, these features are better suited to represent activities in high density videos. We therefore build histogram of motion trajectory based on length and orientation of those trajectories. For this, we represent all the motion trajectories extracted from a video by a 2D feature vector, with 1 dimension each for length and orientation. There are g types of length and h types of orientation, the total combination of length and orientation is $g \times h$. Since, each motion trajectory has a certain length and orientation feature, it can be represented by one of the $g \times h$ combination of length and orientation.

We define Histogram of Trajectories (HoT) for a video, as a 2 dimensional vector of size $k = g \times h$, where each element $E_{i,j}$ represents the number of motion trajectories in the video, with length feature i and orientation feature j, respectively. We consider the entire space of $length\ \times orientation$ as a visual dictionary of the video classes, with each element $E_{i,j}$ in this space as a word in the visual dictionary. So, HoT represents the frequency of occurrence of each word of the dictionary in a video.

We see the visualization of HoT in Figure 3.5. This corresponds to the bag-of-words approach used in many image, video and text classification work found in the literature. For example, consider that we have a video with 100 trajectories, and 10 of those trajectories have length feature p, and orientation feature o. The combination of these properties is represented by a word in the visual dictionary, represented by $E_{p,\,o}$.

Then, the value of $E_{p,\,o}$ is 10, which represents that the video has a feature $E_{p,\,o}$ with 10 magnitudes. HoT for this video is the vector with magnitude for all the features in the visual dictionary. This means, if any video has no feature corresponding to an entry in the dictionary, then those features have a magnitude of zero. Thus, HoT represents the distribution of both length and orientation features of trajectories in a video.

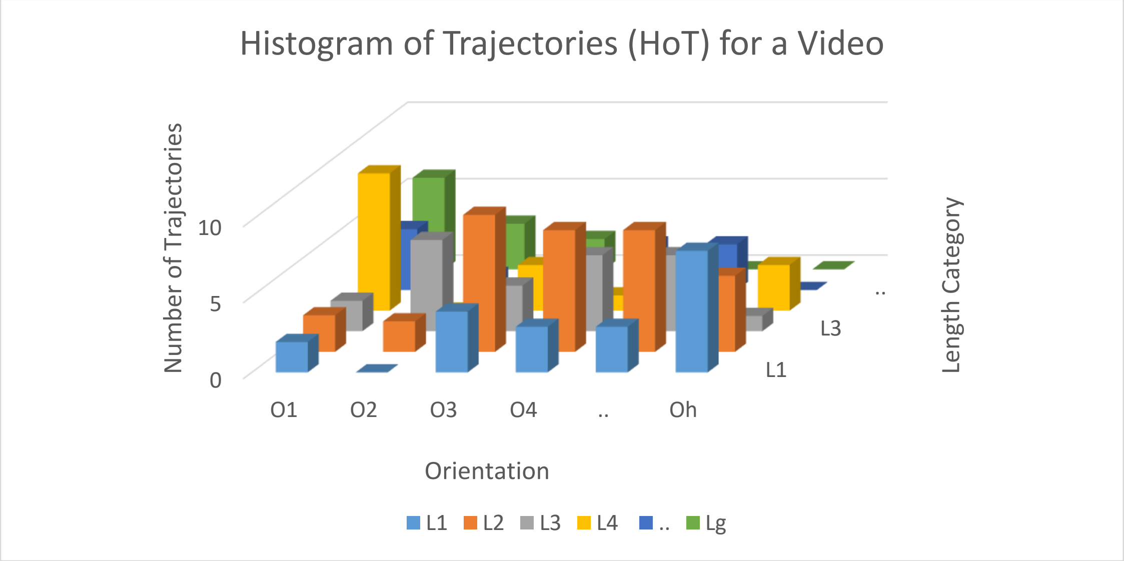

Figure 3.5:Example of HoT feature showing the frequency of occurrence of each trajectory feature in a video

3.5 Trajectory Clustering and Histogram of Trajectory Clusters (HoTC)

In addition to features of individual trajectories, different types of movements can be characterized by motion trajectory clusters. Spatial clustering is widely studied in literature and a number of algorithms are presented and used in practice. We have developed a variant of density based clustering algorithm, DBSCAN [42], for short trajectories, called DBSCAN-ST. DBSCAN is extensively used in density based clustering algorithm with noise detection. In SPS, there is a large number of motion trajectories, throughout the space. The motivation for density based clustering is to find out dominant motion patterns in videos and to detect and remove outlier trajectory from further analysis.

In the next section, we will present a novel approach to cluster short trajectory based on density. We extend the approach for our clustering algorithm from DBSCAN, and use distance measures defined by Lee and Han [8]. We call our algorithm DBSCAN-ST, as it clusters short trajectories which are common in SPS. In Section 3.5.1, we define two distance measure commonly used in measuring similarity between line segments, and in Section 3.5.2 we present DBSCAN-ST.

3.5.1 Trajectory Distance

Measuring the distance between line segments is complicated, mostly because there is no set definition of distance between lines. There is a body of work in pattern recognition with focus on defining robust measure to define distance between line segments [43]. We use two of the distance measure which is also used by Lee and Han [8] in their trajectory clustering work.

Perpendicular Distance ($P_{\perp}$) is the measure of how far away two line segments are from one another. Lee and Han [8] define perpendicular distance between two line segments as the normalized mean between the perpendicular distances of end points of shorter trajectory on the projection of longer trajectory.

Parallel Distance ($P_{//}$) is defined as the minimum distance between the projections of the end points of shorter line segment with end point of longer line segment [8].

The mathematical expression for perpendicular and parallel distance between two line segments is as follows:

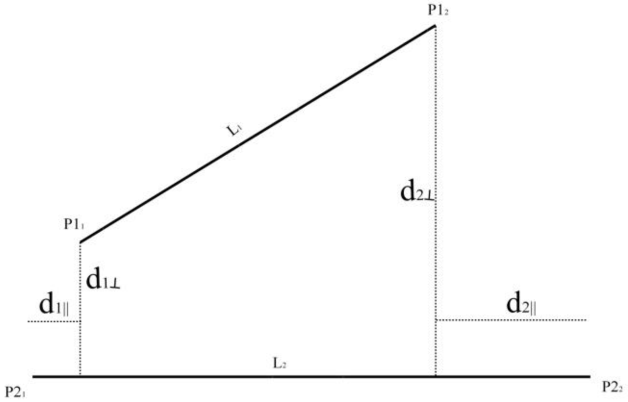

Figure 3.6: Perpendicular and Parallel Distance between two line segments L1 and L2

$$\textit{Perpendicular Distance } (P_{\perp}) = \frac{d_{1\perp}^{2} + d_{2\perp}^{2}}{d_{1\perp} + d_{2\perp}}$$

$$\textit{Parallel Distance } (P_{\|}) = minimum(d_{1\|}, d_{2\|})$$

3.5.2 DBSCAN-ST

Density Based Spatial Clustering of Application including Noise for Short Trajectories (DBSCAN-ST) is a trajectory clustering algorithm we developed, which utilizes approach used in DBSCAN. It uses the distance measures defined in Section 3.5.1 in order to (a) cluster nearby trajectories together and (b) detect and discard noisy or outlier trajectories. However, we approach trajectory clustering with separate threshold measure for parallel and perpendicular distance in order to control the cluster formation with higher granularity. Separate perpendicular and parallel threshold gives the algorithm control on how far away two trajectories can be from each other, and how the length of the trajectories can differ from each other. Finally, as we are dealing with short trajectories, we do not partition trajectories and assume them to be straight. Additionally, density based clustering is helpful in case of dense and crowded video as there is a lot of trajectory fragmentation due to heavy occlusion. This approach enables us to cluster such broken trajectories together. Here we define several key concepts associated with our approach.

- *Perpendicular Distance Threshold ($\epsilon_\perp$)*: Perpendicular Distance Threshold ($\epsilon_\perp$) is the maximum perpendicular distance between two trajectories t_1 and t_2 for them to be in the same neighborhood.
- *Parallel Distance Threshold ($\epsilon_{/\!/}$)*: Parallel Distance Threshold ($\epsilon_{\|}$) is the maximum parallel distance between two trajectories t_1 and t_2 for them to be in the same neighborhood.
- *Minimum Number of Trajectories in a Cluster (N_{min})*: Minimum Number of Trajectories in a Cluster (N_{min}) refers to the minimum number of trajectories in neighborhood, for the trajectories that are not noise.
- *Noise*: The trajectories that do not have at least N_{min} trajectory in their neighborhood are noise.
- *Core Trajectories*: The trajectories that have at least N_{min} trajectory in their neighborhood are called core trajectories. These trajectories are guaranteed to fall in a cluster by the end of the clustering process.

DBSCAN-ST takes four parameters (a) Perpendicular Distance Threshold ($\epsilon_\perp$), (b) Parallel Distance Threshold ($\epsilon_{/\!/}$), (c) Minimum Number of Trajectories in a Cluster (N_{min}) and (d) set of all trajectories, and returns the cluster assignment for each trajectory and a flag signifying if the trajectory is noise or not, as the output.

DBSCAN-ST process can be divided into two distinct section: (a) pre-processing and distance calculation and (b) clustering. In pre-processing, we calculate the pair wise perpendicular and parallel distance between all trajectories using the algorithm Pairwise_Distance presented in Figure 3.9. Pairwise_Distance iterates through each unique pair of trajectories, and calls Parallel_Distance and Perpendicular_Distance functions to calculate the parallel and perpendicular distance, respectively, between them.

Then, for clustering, we visit each trajectory (t_i) and its neighborhood. We add the trajectory to a cluster (C_i) if it has at least N_{min} neighbors, marked the trajectory as visited and systematically add all the neighbors to C_i. Finally, we continue to process each neighbor (Nt_i) of t_i by recursively adding all trajectories which are not visited and have at least N_{min} neighbors. When there are no more trajectories for current cluster, we pick an unexplored trajectory, update the cluster number and repeat the above given process. The algorithm for this process is described in Figure 3.10, with supporting algorithms in Figure 3.7, 3.8 and 3.9.

```
Function Parallel_Distance (Trajectory $t_i$, Trajectory $t_j$)

Input: Two Trajectories $t_i$ and $t_j$
Output: Parallel Distance ($d_{i,j\parallel}$) between $t_i$ and $t_j$

Begin
    Find the length of $t_i$ and $t_j$, determine longer ($L$) and shorter($S$)
    trajectories.
    Project end points of $S$ i.e.$E_{S1}$ and $E_{S2}$ on $L$ as $E'_{S1}$ and $E'_{S2}$. Let end
    points for $L$ is $E_{L1}$ and $E_{L2}$
    Find the distance between $E_{L1}$ and $E'_{S1}$ as $d_{1\parallel}$ and $E_{L2}$ and $E'_{S2}$ as $d_{2\parallel}$
    Return $d_{ij\parallel}$=minimum ($d_{1\parallel}$,$d_{2\parallel}$)
End
```

Figure 3.7: Algorithm to calculate parallel distance between two trajectories

```
Function Perpendicular_Distance (Trajectory $t_i$, Trajectory $t_j$ )

Input: Two Trajectories $t_i$ and $t_j$
Output: Perpendicular Distance ($d_{i,j\perp}$) between $t_i$ and $t_j$

Begin
    Find the length of $t_i$ and $t_j$, determine longer ($L$) and shorter($S$) trajectories.
    Project end points of $S$ i.e.$E_1$and $E_2$ on $L$ as $E_1'$ and $E_2'$
    Find the distance between $E_1$ and $E_1'$ as $d_{1\perp}$ and $E_2$ and $E_2'$ as $d_{2\perp}$
    Return $d_{i,j\perp} = \frac{d_{1\perp}^2 + d_{2\perp}^2}{d_{1\perp} + d_{2\perp}}$
End
```

Figure 3.8: Algorithm to calculate perpendicular distance between two trajectories

```
Function Pairwise_Distance (Trajectory Array T [N])

Input: Array of all trajectories in a video $T = [t_1, t_2 \dots t_N]$
Output: $N \times N$ distance matrices $d_\perp$ and $d_\parallel$ representing pairwise perpendicular and
parallel distances respectively between all $N$ trajectories.

Begin
    $D_\perp = [], D_\parallel = []$
    for $i=1$ to $N$
        for $j=1$ to $N$
            if $i==j$
                $D_\perp [i] [j] = 0, D_\parallel [i] [j] = 0$
            else
                $D_\perp [i] [j] = Perpendicular\_Distance (t_i, t_j)$
                $D_\parallel [i] [j] = Parallel\_Distance (t_i, t_j)$
            end if
        end for
    end for
    return $D_\perp, D_\parallel$
End
```

Figure 3.9: Algorithm to calculate pairwise perpendicular and parallel distance matrices between all trajectories

```
Function DBSCAN-ST ($T[N], \epsilon_{\perp}, \epsilon_{\parallel}, N_{min}$)

Input: Array of all trajectories in a video $T = [T_1, T_2 \ldots T_N]$
       Perpendicular Distance Threshold ($\epsilon_{\perp}$)
       Perpendicular Distance Threshold ($\epsilon_{\parallel}$)
       Minimum Number of Trajectories in a Cluster ($N_{min}$)

Output: ClusterAssignment $[N] = [C_i, C_j \ldots C_M]$ cluster assignment for each trajectory
        NoiseFlag $[N]$, where True means corresponding trajectory is a Noise and
            vice-versa
Begin
    $PWD_{\perp}, PWD_{\parallel}$ = Pairwise_ Distance (T);
    Visited = False [N], NoiseFlag = False [N], CA[N], i=0, CNum= 1
    while all trajectories are not visited
        if   Visited[i] == False
            Visited [i] = True
            Neighbors = Get_Neighbors($t_i, \epsilon_{\perp}, \epsilon_{\parallel}$)
            if size (Neighbors) < $N_{min}$
                NoiseFlag [i] = True
            else
                CA[i] = CNum
                for all neighbors[k] of Trajectory T[i]
                    CA [k]= CNum
                    Visited[k] = True
                end for
            end if
        end if
    end while
End
```

Figure 3.10: DBSCAN-ST algorithm

Complexity Analysis: DBSCAN-ST is asymptotically similar to DBSCAN, and hence has similar runtime performance. The runtime is dependent on total number of trajectories being clustered. The most time consuming operation for this is Pairwise_Distance which calculates perpendicular and parallel distance between each unique trajectory pair. So, for the number of trajectories N, runtime complexity is given by $O(N^2)$.

We can see from the algorithm that, the space requirement for DBSCAN-ST completely depends on the pairwise distance between all trajectories, since it requires 2 different $N \times N$ matrices to store parallel and perpendicular distances between all trajectory pairs. Hence, the space complexity for DBSCAN-ST is $O\left(N^2\right)$.

3.5.3 Histogram of Trajectory Clusters (HoTC)

Since we use HoT features to represent the activities in entire videos, we extend that approach to trajectory clusters as well. As our trajectory clusters are spatially divided, we assume that different clusters formed by using DBSCAN-ST on motion trajectories can represent different activities. So, clustering makes the classification process more granular and improves efficacy in cases where there are more than one distinct classes within a video.

It is plausible that not all spectators in a SPS event are doing the same thing. For example, spectators watching sports can be supporters of different teams, in different region within spectator space. For video that contain supporters from both team, it is essential to evaluate those spectators differently, as they might be expressing opposing emotion. Clustering allows us to analyze spectators with more granularity, and detect multiple classes of action within a same video.

Consequently, we process each trajectory cluster separately, by applying the same method applied to entire video in the HoT approach. This allows us to form several histograms of trajectories for each video, depending on the number of clusters in the video. Hence, we call this approach histogram of trajectory clusters.

3.6 Classification

The last step in our approach is to classify a video into a set of predefined classes. We use a machine learning approach for this step. Many different classifiers are proposed in literature and new ones are being presented continuously. The classifier function is trained with HoT and HoTC features, respectively, from training dataset to create trained classifiers. These classifiers can then be tested with HoT and HoTC features of testing dataset. For our research, we have examined the performance of the following classification approaches.

- Support Vector Machines (SVM): SVM is a classifier that is defined by a separating hyperplane [44]. Some of the most widely used video classification algorithms use non-linear support vector machine (SVM) for video classification as SVM can handle data with higher dimensionality, and is less prone to exhibit multiple local minima and over-fitting. SVM is also found to be a better classifier for video data, when the videos are highly occluded or contain high level of variation in illumination [45].
- NaïveBayes Classifier: NaïveBayes is a simple, probabilistic classifier based on the assumption that all features are independent of one another [46]. We expect the different classes of SPS videos to have significantly different motion trajectory orientation and length based on the activities in those classes. So we test classification with NaïveBayes as strong variability in motion trajectories based on classes of videos could result in classifier with high efficacy.
- Ensemble based classifiers: We also use majority voting, which is an ensemble based approach, with four different classifiers i.e. NaïveBayes, BayesNet, SVM

and J48 decision tree. This approach assign class to a video based on majority vote i.e. all four classifier classify the video into different classes, and the video is assigned class with the highest votes. In some cases, ensemble based classification has shown to improve classification efficacy when individual classifiers perform weakly [47].

- Deep Learning: Deep learning neural networks are special type of machine learning networks, containing convolutional layers, and deep architecture [48]. In video and image classification, deep learning based architecture is used to build holistic classification models which takes raw video or images as input dataset and create a classification model directly based on those input. We use deep learning based classifier which is proposed by Tran et al [26], to classify our videos into different classes.

With the design of our research, once our features (HoT and HoTC) are built, any classifier model can be trained with those features, and used to classify videos. Those features are independent of any classification model, and thus can be used on any new classification technique.

Chapter 4

Implementation and Result

In this chapter, we evaluate the performance of algorithms and a comparison of HoT and HoTC on a set of real life SPS videos. First, we discuss about the dataset used in our research, including the segmentation and labelling process. Second, we will present the length and orientation analysis of the motion trajectory of all categories of video in our dataset. Third, we will also discuss the results from several experiments, to classify SPS video using HoT, HoTC and deep learning. And finally, we discuss the outcome of all the classification experiments.

4.1 Dataset

We created the dataset for our research from four surveillance videos of different college football games played by the UNL football team at home, during 2015-2016 season. These videos were all captured using a fixed camera and fixed zoom, with specifications described in Table 4.1. Snapshots of these videos are given in Figure 4.1 and Figure 4.2, respectively.

Table 4.1:Game video details

Video-id	Duration	Resolution	Frame Per Second
Day Game 1	4.5 hours	1920 by 960	25
Day Game 2	4 hours	1920 by 960	25
Night Game 1	5 hours	1920 by 960	25
Night Game 2	4 hours	1920 by 960	25

Figure 4.1: An image from a fixed camera surveillance video for a night game

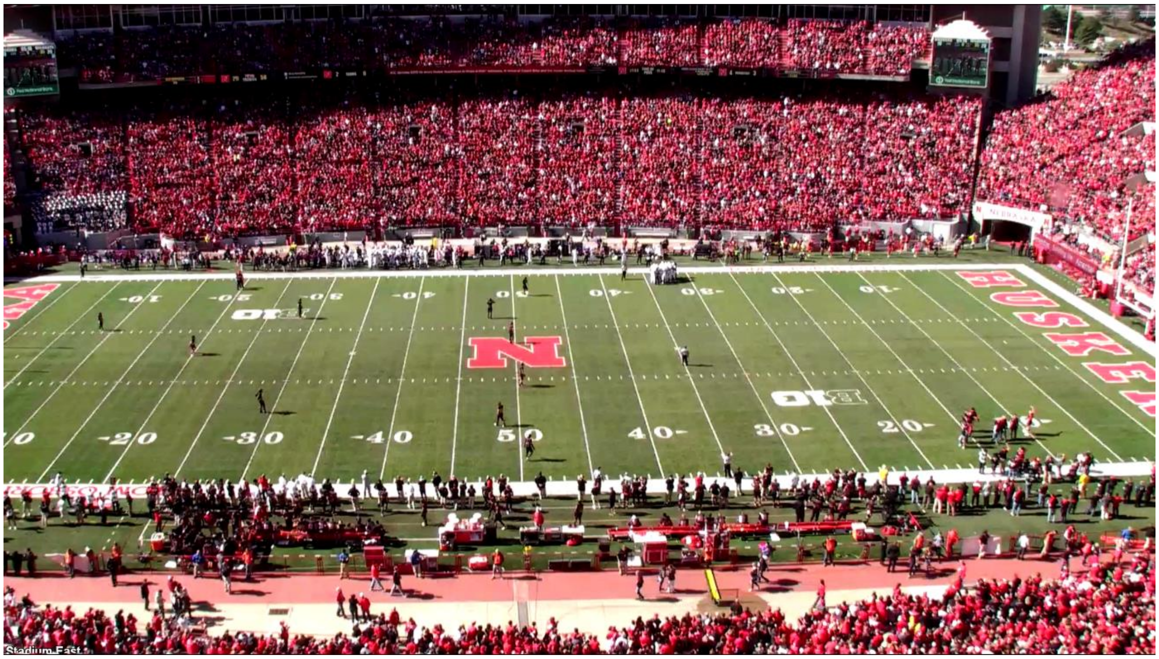

Figure 4.2: An image of a fixed camera surveillance video for a day game

4.1.1 Performer Spectator Segmentation

We segmented the videos spatially into performer and spectator spaces manually. Since the camera was fixed and had a fixed zoom, it was accomplished with relative ease. In the general case, the space can be segmented dynamically using properties of spectators and performers as described in Section 1.2. At the end of this step, we had 4 videos each for the spectator space and performer space. The stadium region represented the *spectator space* and play field region is considered as the *performer space*. Snapshots from both categories of video are presented in Figure 4.3 and Figure 4.4, respectively. Space that is neither spectator nor performer is removed from further analysis.

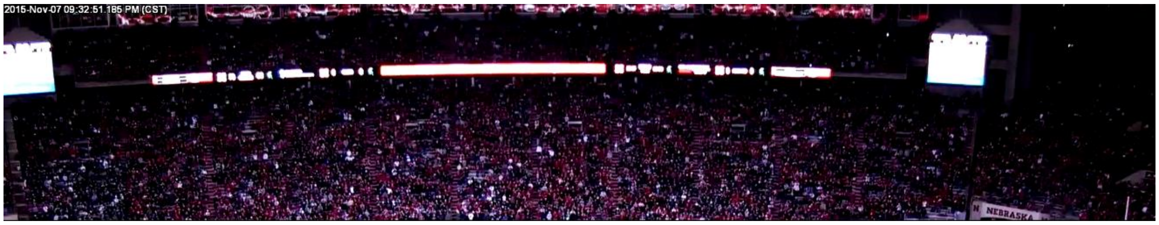

Figure 4.3: Snapshot of Spectator Video

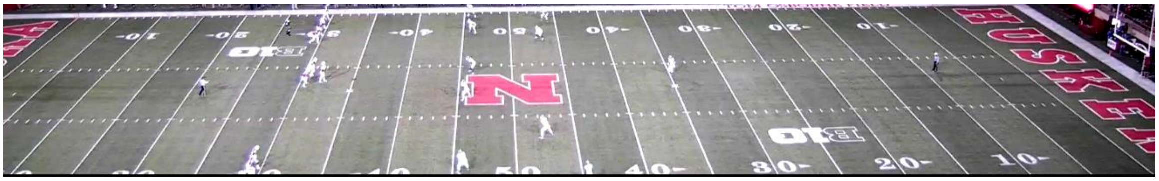

Figure 4.4: Snapshot of Performer Video

4.1.2 Temporal Segmentation

We then divided the four videos into smaller segments to build a dataset for training and evaluation. For simplicity, we divided each video into a fixed length of 5 seconds. Ideally, we would have liked to segment the video corresponding to episodes of activity. However, this is a complex process in the general case, because the behavior of crowd is non-deterministic. The challenges in fixed length segmentation are: (a) single event may be broken into multiple segments, (b) a single segment may have multiple activities.

4.1.3 Activity Classes and Ground Truth Development

We found spectator crowds to be very dynamic and show a wide range of emotion, as they are observing some form of performance or involved in religious or political gathering. Spectator emotion and response evolved based on the activities of the performers/players. This behavior can be observed commonly spectators from many other SPS videos. For example, sports spectators get excited or dejected based on how their teams perform, and these emotions and response evolve continuously. Musical concert spectators are show high level of excitement for performance of popular songs by performers. Sometimes, sports spectators turn unruly or violent during the course of the game if the results do not go well. It is apparent from observing all kinds of spectator crowds that excitement is an important aspect of a crowd behavior. We, therefore, define three classes of behavior for our classification.

- *Active:* Spectators were excited or happy for majority of the time.
- *Passive:* Spectators were not showing any overt excitement and were generally calm for majority of the time.
- *Mixed:* Spectators were active or passive for roughly equal amount of time.

Performers, on the other hand, either do their primary performance, or are in a resting state. This characteristic is common for performers throughout the entire SPS. For example, musical performers often take short amount of rest in between their musical performances, sportspersons take short to long period of rest during a game. Although the length of rest period varies, performers are perpetually in one of these two states. Therefore, for the performer video, we define the following classes:

- *Play*: A play was being made for the majority duration.
- *No-play:* No play was being made for the majority duration.
- *Mixed*: Roughly equal time of play and no-play period.

In total, 3500 spectator videos generated from four different games were manually classified as active, passive or mixed. Similarly, 3500 performer videos classified as play, no-play or mixed. This gave us a significant amount of data for our experimentation. We selected 1000 videos for each activity class, from each category for our experimental analysis. So, we had 1000 each for play, no-play and mixed classes of performer video and 1000 each for active, passive and mixed classes of spectator videos.

4.2 Hardware and Software Configuration

Since we present the comparison of our classification technique with deep learning, all of our experiments were done in the same system. We used a computer system with 60 GB of memory, 8 processor cores running Ubuntu Cloud 14.04 LTS operating system.

We used C3D, which is a Caffe [49] based deep learning library developed by Facebook research for deep learning based video classification. This is because we found C3D to be the best deep learning based video classifier library available today [26]. It is important to note that Caffe is an open source deep learning library, built on C/C++ and consequently faster in terms of instruction execution than other environment like Matlab. For Histogram of Trajectories (HoT) and Histogram of Trajectory Clusters (HoTC), we use a combination of C++ and Matlab. Trajectory extraction for both HoT and HoTC was implemented in C++ whereas, pre-processing, quantization, training and testing were done on Matlab.

4.3 Analysis of Spectator and Performer Videos

In Section 3.4.1 and 3.4.2, we defined the length and orientation properties of trajectories, respectively, derived from SPS videos. We now present the analysis of spectator and performer videos with respect to length and orientation in Section 4.3.1 and Section 4.3.2, respectively.

4.3.1 Spectator Space

As discussed in Section 4.1 spectator videos are classified into three categories: active, passive and mixed. Each of these categories of videos have different motion trajectories property with respect to length and orientation, which are presented next.

Before that, we provide analysis to compute the length and orientation features, which were presented in section 3.4.1 and 3.4.2, respectively. For orientation feature, we choose the number of orientation-category (g) as 9, based on HOG [14]. Then, we calculate the direction of the orientation using Equation 3.2. and calculate the orientation using Equation 3.3. The 9 orientation possible from this equation are given in Table 4.2.

Table 4.2: Orientation Value mapping to corresponding trajectory direction

Orientation	**Range of Angle (Degrees) Direction**
1	0-20
2	20-40
3	40-60
4	60-80
5	80-100
6	100-120
7	120-140
8	140-160
9	160-180

Similarly, the length for a trajectory is computed using Equation 3.3 in Section 3.4.2. Then, using Equation 3.4, we divide the trajectories into three classes based on their length. Table 4.3 shows the distribution of trajectory lengths. We choose the number of length-category h as 3. Finally, based on the distribution given in Table 4.3, we divide the trajectories into three classes: *short* (0-2], *medium* (2-4], and *long* (>4).

Table 4.3: Average trajectory length distribution for spectator videos, 1 from each category

	Average Number of Trajectories
<1	7298
1-2	6831
2-3	9996
3-4	9786
4-5	7832
>5	9716

After we have both length and orientation features, we analyze the all three categories of spectator videos in terms on those two features. Next, we provide those analyses.

Active: Active spectators are generally excited about the performance due to various reason: the team they support could be winning, or the artist they are watching could be giving an amazing performance and we see these excitements in their motion. They are animated, and this shows in their movement, as they cheer and applause in expressively. We observe that active crowd have more vertical motion than horizontal from Figure 4.5. From Figure 4.6, we observe that long and medium trajectories are more common than short trajectories.

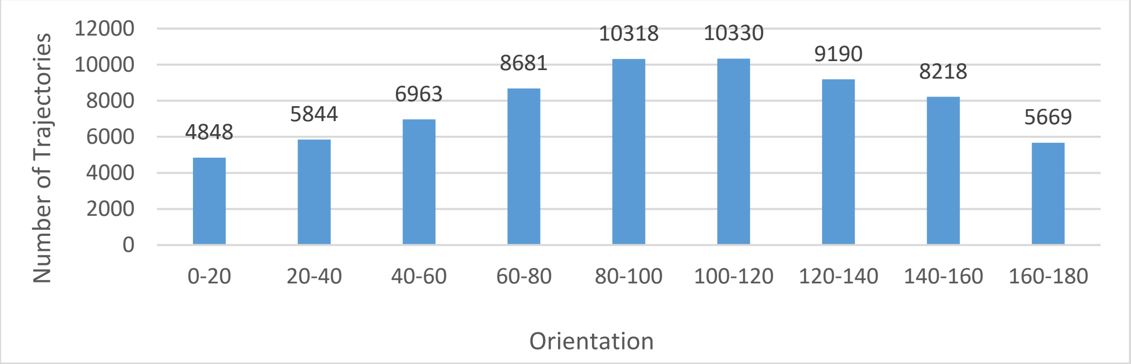

Figure 4.5: Trajectory distribution with respect to orientation in active spectator videos

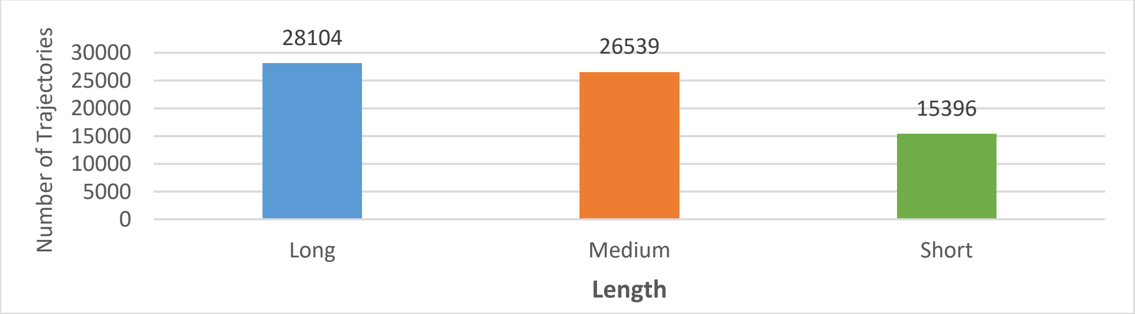

Figure 4.6: Trajectory distribution with respect to length of active spectator video

Passive: Passive spectators are not happy or excited about the performance, and this effects the amount and type of motion trajectories they generate. Overall, passive spectators have less number of trajectories, with more horizontal motion than vertical. They are not involved in activities such as cheering and waving, which reduces the overall number of trajectories. The horizontal trajectories of passive spectators can be attributed to their movement out their seats in order to go out of the stadium for breaks, half time etc. Some of the vertical trajectories are because of the limited amount of cheering from the spectators as well as their movements in the vertical aisles. In Figure 4.7, the trajectory orientation distribution for passive spectators is presented.

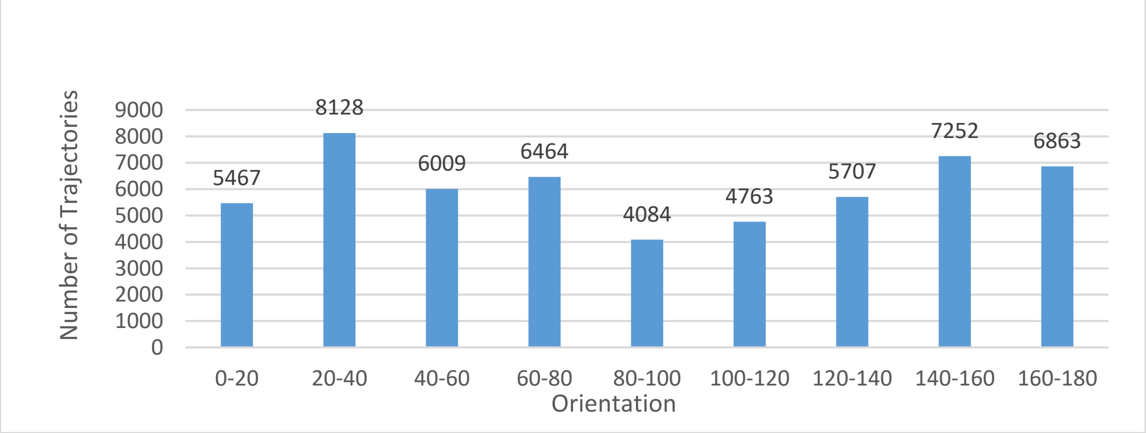

Figure 4.7: Trajectory distribution with respect to orientation in passive spectator videos

In terms of length, both active and passive videos have similar trajectories, with higher number of longer trajectories than longer trajectory. Although the relative distribution of trajectory length is similar, it is important to note that the overall number of trajectory are higher in active video than in passive video. We see from Figure 4.8 that passive spectator video has shorter motion trajectories than longer motion trajectories.

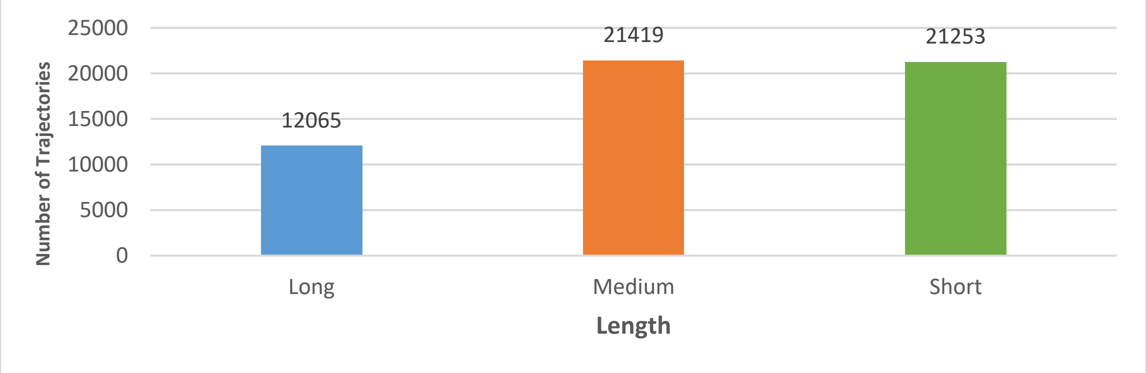

Figure 4.8: Trajectory distribution with respect to Length of passive spectator video

Mixed: Similarly, mixed performers have about average trajectory distribution in both length and orientation compared to Active and Passive performer video. This is because mixed videos contain roughly equal amount of active and passive region and duration. The trajectory distribution statistics is presented in Figure 4.9 and Figure 4.10.

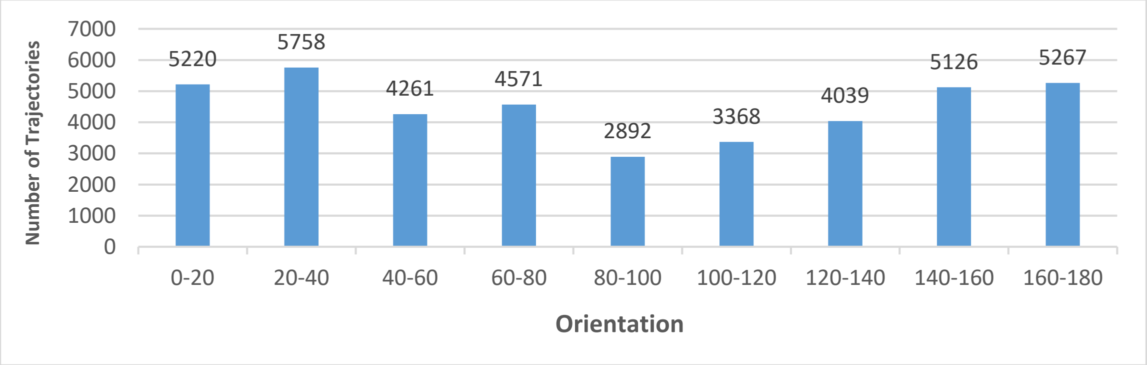

Figure 4.9: Trajectory distribution with respect to orientation in mixed spectator videos

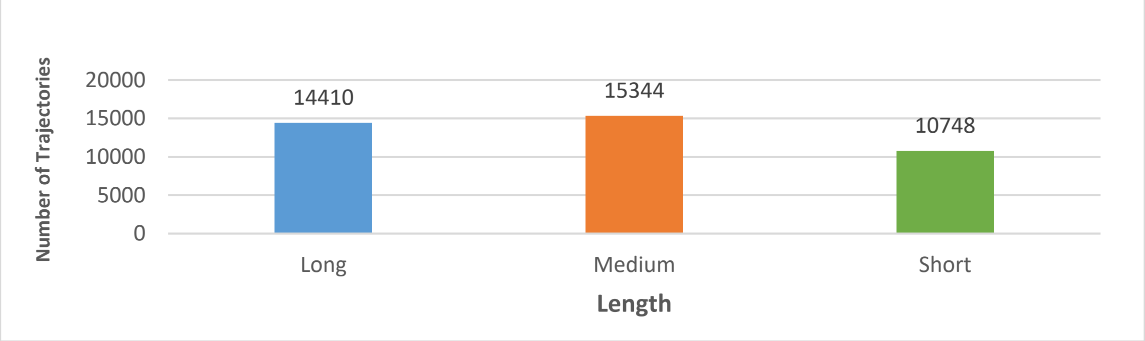

Figure 4.10: Trajectory distribution with respect to length of mixed spectator video

4.3.2 Performer Space

The orientation and length feature for performer space is calculated exactly same as the spectator space. We choose same 9 orientations as spectator videos whereas for trajectories, we choose 3 length category, similar to spectator videos, based on Table 4.4, which are: *short* (0-5], *medium* (5-10], and *long* (>10).

After we have both length and orientation features, we analyze the all three categories of spectator videos, play, no play and mixed, in terms on those two features. Next, we provide those analyses.

Table 4.4: Average trajectory length distribution for performer videos, 1 from each category

Length Range (units)	Number of Trajectories
0-1	1048
1-2	1191
2-3	898
3-4	1296
4-5	772
5-6	1166
6-7	1416
7-8	9000
8-9	998
9-10	1036
10-11	1011
11-12	929
12-13	932
13-14	980
14-15	772
>15	630

Play: In play videos, we expect the players to run horizontally, for both offense and defense. Although there can be several types of play, and not all plays horizontal, majority of the movement must be horizontal. This is presented in the trajectory distribution graph in Figure 4.11. We see that the number of trajectories slowly decrease as the orientation becomes more vertical i.e. increases from 0 towards 90 and increases again when the orientation becomes horizontal i.e. increases from 90 towards 180. In addition, we see from the same figure that for play videos, more trajectories are horizontal (0 to 45 and 135 to 180) than vertical (45 to 135 degrees).

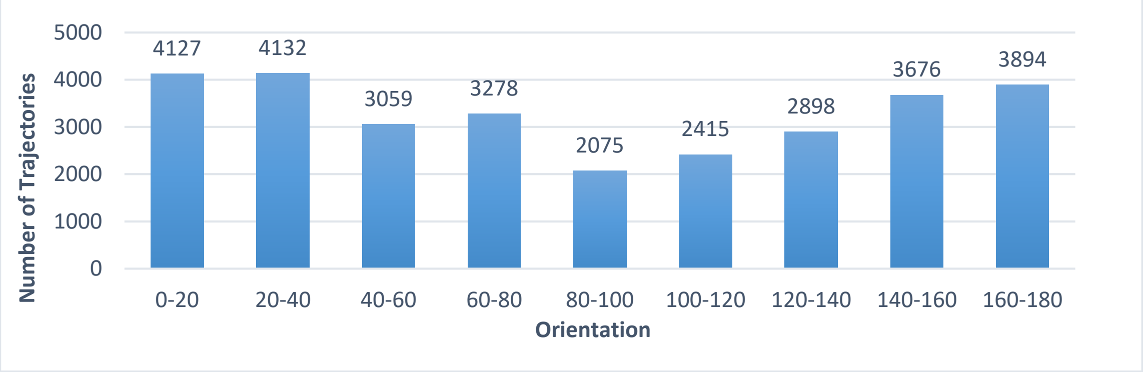

Figure 4.11: Trajectory Distribution with respect to orientation in play performer video

In addition, from Figure 4.12 we see that the trajectory distribution with respect to length also changes as the trajectory length decreases. The number of trajectories are higher in long than in medium and higher in medium than in short trajectories i.e. there are higher number of longer trajectories than shorter trajectories.

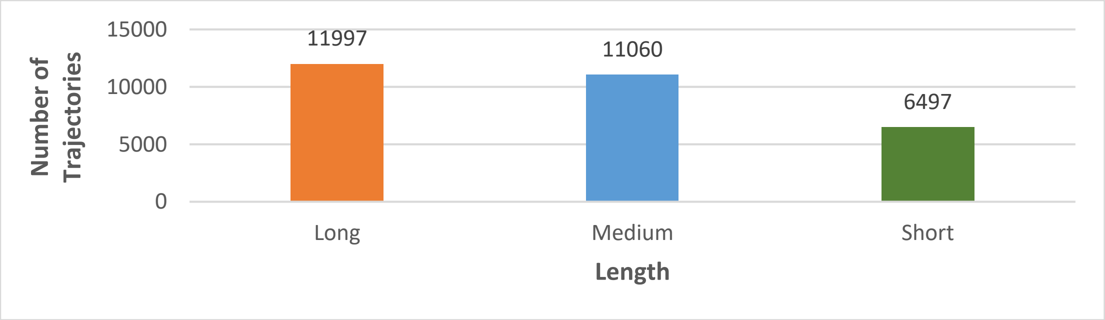

Figure 4.12: Trajectory distribution with respect to length of play performer video

No Play: For no play video, we expect to see more vertical movement between the sidelines and the field. We expect the players to run towards the sidelines for timeouts, offense-defense switch, and rest, which results in a more vertical motion than horizontal. This trend is apparent from Figure 4.13 as we can see maximum trajectories in the 100°-120° bucket, and higher number of vertical trajectories than horizontal trajectories.

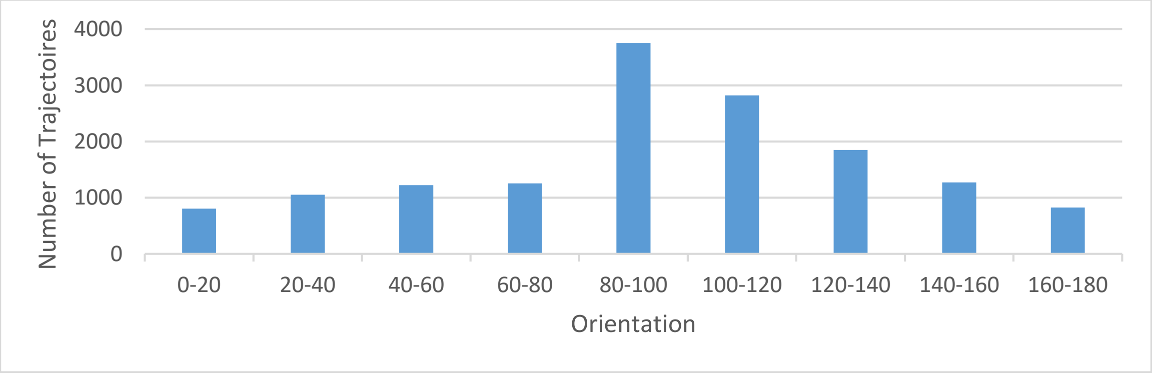

Figure 4.13: Trajectory distribution with respect to orientation in no play performer video

Moreover, with respect to length of the trajectory, we observed that there are high number of short trajectories than long trajectories. This is also an expected observation, as no play video generally will have high number of short movements from field to the sidelines and low number of longer movement by the players. This is observed in Figure 4.14, where we see that the number of short trajectories is more than 3 times the number of long trajectories.

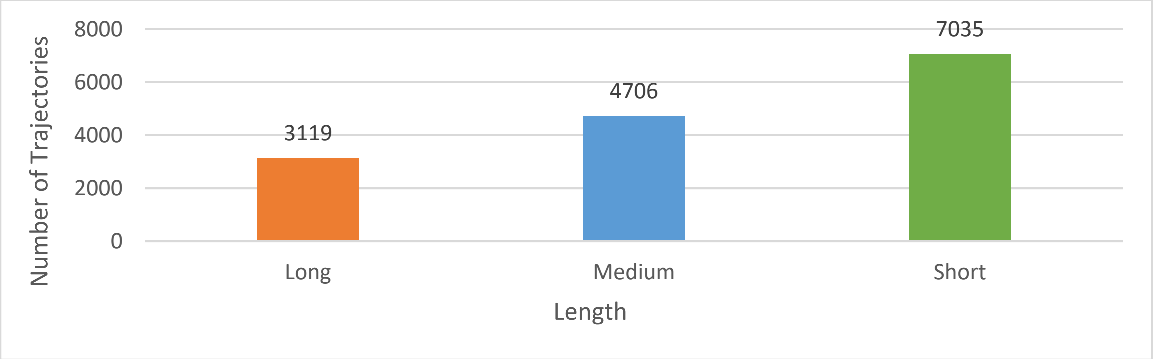

Figure 4.14: Trajectory distribution with respect to length of no play performer video

Mixed: Similarly, for performer video of mixed category, we see a uniform distribution of trajectories with respect to orientation. We can see from Figure 4.15 that, overall, the distribution of trajectories is uniform for all orientation. This differentiates mixed videos from both play and no play video. This distribution is also expected as mixed video contain mixture of play and no play situation, and so the trajectory distribution averages between that of play and no play videos.

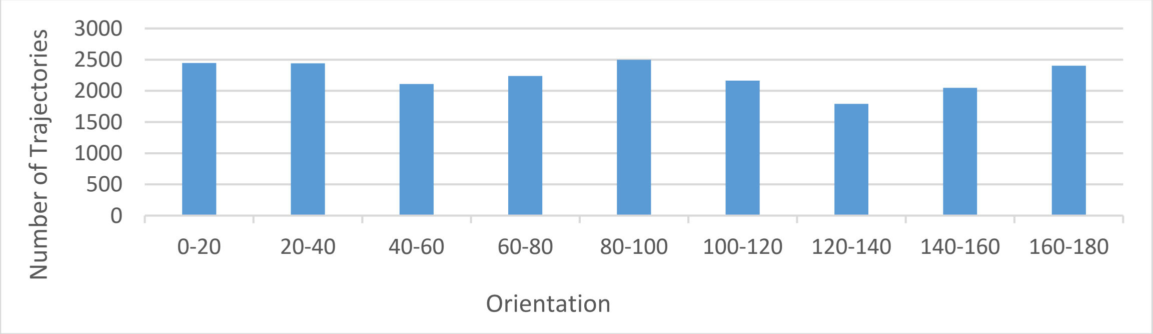

Figure 4.15: Trajectory Distribution with respect to orientation in mixed performer video

Length wise, we again see more uniform distribution, as there are similar number of long, medium and short trajectories. This is again because as mixed video class has behavior of both play and no play class, the number of trajectory which are long, medium and short averages out to similar quantity. This behavior can be observed in Figure 4.16.

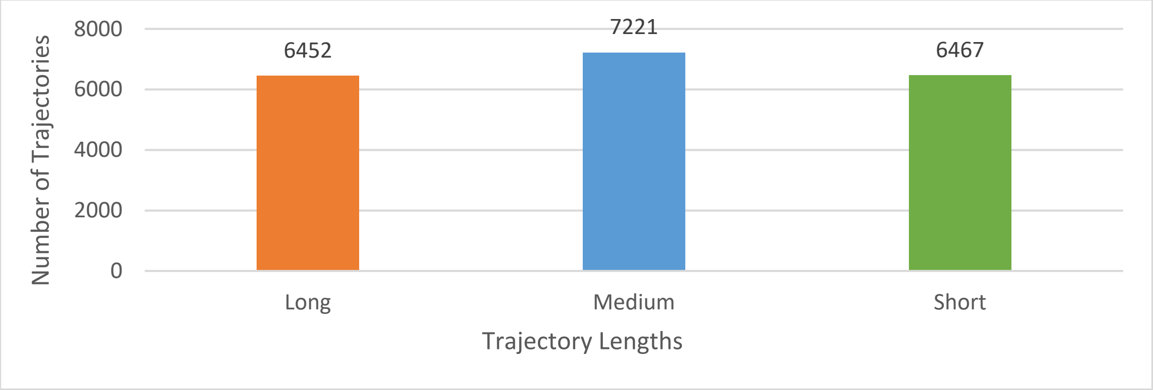

Figure 4.16: Trajectory distribution with respect to length of mixed performer video

4.4 Comparison of Video Classes

We perform statistical test on the trajectory distribution between spectator and performer videos as well as videos of all categories in both space. We present the result of spectator-performer comparison in Section 4.4.1 and comparison between different categories in each space in Section 4.4.2.

4.4.1 Spectator and Performer Space

We see from the analysis in Section 4.3.2 and Section 4.3.2, that the overall, there are more motion trajectories in spectator space than in performer space. This is because the spectator space, although dense, has many actors/people. This results in higher number of people moving in different directions, giving spectator scenes lot more motion than performer scenes. We perform variance analysis with 95% confidence and find this result to statistically significant i.e. #trajectories in spectator space > #trajectories in performer space.

4.4.2 Action Categories in Spectator and Performer Space

We performed variance analysis (with 95% confidence) to evaluate if the difference in distribution of motion trajectories between active, passive and mixed classes of spectator videos were statistically significant or not. Part of the result from these tests are given here whereas detailed results from this test are given in Table A.1.1 through A.1.4 in the appendix. As discussed previously, the three classes of videos have different values for length and orientation. The length and orientation distribution between active, passive and mixed videos were statistically significantly different from one another in almost all category of length and orientation. We can see part of the result in Table 4.5 and Table 4.6.

Table 4.5: Result of variance analysis on active vs passive vs mixed spectator videos trajectory distribution with respect to length. #active, #passive and #mixed are number of trajectories in active, passive and mixed video respectively of length corresponding to the value given in their respective rows. Confidence: 95%

Length	Alternative Hypothesis	Significance
Long	#active > #passive	Yes
Medium	#active > #passive	Yes
Short	#passive > #active	Yes
Long	#active > #mixed	Yes
Medium	#active > #mixed	Yes
Short	#mixed > #active	Yes
Long	#mixed > #passive	Not Statistically Significant
Medium	#mixed > #passive	Yes
Short	#mixed > #passive	Yes

Table 4.6: Result of variance analysis on active vs passive spectator videos trajectory distribution with respect to orientation. #active and #passive are number of trajectories in active and passive video respectively of orientation corresponding to the value given in their respective rows. Confidence:95%

Orientation	Alternative Hypothesis	Significance
0-20	#active > #passive	Yes
20-40	#active > #passive	Yes
40-60	#active > #passive	Yes
60-80	#active > #passive	Yes
80-100	#active > #passive	Not Statistically Significant
100-120	#active > #passive	Not Statistically Significant
120-140	#active > #passive	Yes
140-160	#active > #passive	Yes
160-180	#active > #passive	Yes

Similarly, we performed variance analysis (with 95% confidence) to evaluate if the difference in distribution of motion trajectories between play, no play and mixed classes of performer videos were statistically significant or not. We present part of the result in Table 4.7 and Table 4.8, with all other results in Appendix A, in Table A.2.1 to A.2.4. We see from Table 4.7 and Table 4.8 that motion trajectories of play, no play and mixed spectator videos are different from one another with statistical significance in most cases.

Table 4.7: Result of variance analysis on play vs no play vs mixed performer videos trajectory distribution with respect to length. #play, #no play and #mixed are number of trajectories in play, no play and mixed video respectively of length corresponding to the length parameter given in their respective rows. Confidence:95%.

Length	Alternative Hypothesis	Significance
Long	#play > #no play	Yes
Medium	#play > #no play	Yes
Short	#no play > #play	Yes
Long	#play > #mixed	Yes
Medium	#play > #mixed	Yes
Short	#mixed > #play	Yes
Long	#mixed > #no-play	Not Statistically Significant
Medium	#mixed > #no-play	Yes
Short	#mixed > #no-play	Yes

Table 4.8: Result of variance analysis on play vs no play performer videos trajectory distribution with respect to orientation. #play and #no play are number of trajectories in play and no play video respectively of orientation corresponding to the value given in their respective rows. Confidence: 95%

Orientation	Alternative Hypothesis	Significance
0-20	#play > #no play	Yes
20-40	#play > #no play	Yes
40-60	#play > #no play	Yes
60-80	#play > #no play	Yes
80-100	#no play > #play	Yes
100-120	#play > #no play	Not Statistically Significant
120-140	#play > #no play	Yes
140-160	#play > #no play	Yes
160-180	#play > #no play	Yes

4.5 Efficacy Comparison of Different Video Classification Techniques

In a preliminary evaluation of classifiers, we trained using a subset (100 from each class) of videos with a large number of classifiers including SVM, Random Forest, J48 Decision Trees, NaïveBayes and BayesNet. The motivation was to determine the best classifier for detailed experiments and comparison of the two kinds of features proposed in our research. Table 4.9 shows the performance of the top three approaches. We performed preliminary experiments on spectator and performer videos, using HoT and HoTC, to compare the efficacy of the classification techniques, and determine the best classifier for comprehensive experimentations. We ran the classification on 100 videos each from spectator and performer spaces for HoT, and evaluated the classification accuracy, using (a) SVM, (b) NaïveBayes (c) Majority Voting Ensemble Classifier (using NaïveBayes, SVM, J48 and BayesNet). The comparison on this accuracy is presented in Table 5 below. Based on this experiment, we selected that Naïve Bayes approach for rest of the experiments.

Table 4.9: Comparison of efficacy of different classifiers

Classifiers	Spectator		Performer	
Features	HoT	HoTC	HoT	HoTC
NaïveBayes	84.18%	84.33%	88.33%	87.46%
SVM	80.33%	82.56%	85.67%	83.67%
Majority Voting	81.67%	80.23%	78.53%	81.79%

We see from Table 4.6 that, in both spaces, NaïveBayes out performs other classification techniques. We attribute this to the strong differences in HoT and HoTC features between the different categories of videos in spectator and performer spaces. For our comprehensive experiments and comparison with deep learning based classification, we used NaïveBayes classifier.

4.6 Classification in Spectator Space

We trained a NaïveBayes classifier with 1500 fixed camera spectator videos, and tested the classifier with another 1500 video. The NaïveBayes classifier yielded 85.2% correct classification; the confusion matrix for this experiment as shown in Table 4.10.

Table 4.10: Confusion matrix - NaïveBayes classification for spectator video

	Active	**Passive**	**Mixed**	**Class**
Active	426	32	42	85.2%
Passive	51	433	16	86.6%
Mixed	60	21	419	73.25%
Class Recall	79.93%	89.09%	87.84%	

One of the reason for incorrect classification of videos was the inherent bias of a single observer in ground truth generation. As those videos were manually classified as one of the three categories by a single observer, the class of the videos are subjective to the observers' opinion.

In addition, we also ran 10-folds cross validation, with the entire 3000 video dataset i.e. we created 10 partition of the data set, used 9 partitions to train, and 1 partition to test, for 10 times, each with different set of training and testing partitions. We present the cross-validated result below in Table 4.11, which has the average accuracy of 87.7%. We also observe from Table 4.11 that the prediction efficacy of our classifier is consistent throughout the 10-folds cross validation. Hence, the HoT features that we use to classify has the capability to train effective classifier and it trains classifier models independent to the training set.

Table 4.11: Cross validation result for spectator video classification, correct prediction shaded with green

Iteration #	**Active**			**Passive**			**Mixed**		
	Active	Passive	Mixed	Active	Passive	Mixed	Active	Passive	Mixed
1	77	3	4	6	78	4	12	9	107
2	89	2	5	4	81	4	10	7	98
3	98	6	5	8	86	9	4	8	76
4	90	4	4	9	92	5	3	6	87
5	77	2	4	10	99	11	8	4	85
6	90	8	2	11	93	1	4	5	86
7	105	9	5	3	71	6	8	5	89
8	91	2	5	12	78	11	14	4	83
9	89	2	6	9	92	2	7	11	82
10	99	11	6	5	94	7	10	8	60

4.7 Classification in Performer Space

Again, we trained a classifier of spectator video into active, passive and mixed classes NaïveBayes classifier with 1500 fixed camera performer videos, and tested with another 1500 videos. Finally, we validated the result with 10-folds cross validation. The classifier performed with 89.6% accuracy, which is better than classification of spectator videos. We present the confusion matrix for this classification in Table 4.12.

Table 4.12: Confusion matrix - NaïveBayes classification for performer video

	Play	**No Play**	**Mixed**	**Class**
Play	448	14	38	89.6%
No Play	30	458	12	91.6%
Mixed	40	25	435	87%
Class Recall	84.84%	92.15%	89.69%	

We present a snapshot of both correctly classified and misclassified performer videos in Figure 4.17 and Figure 4.18, respectively. Similar to spectator videos, some misclassification can be attributed to the inherent bias in the manual classification of the testing data set by the observer.

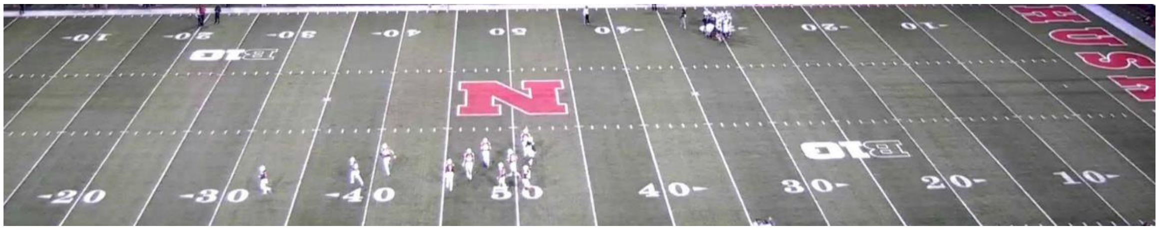

Figure 4.17:Snapshot of video correctly classified as no play. The players are moving in position to make a play.

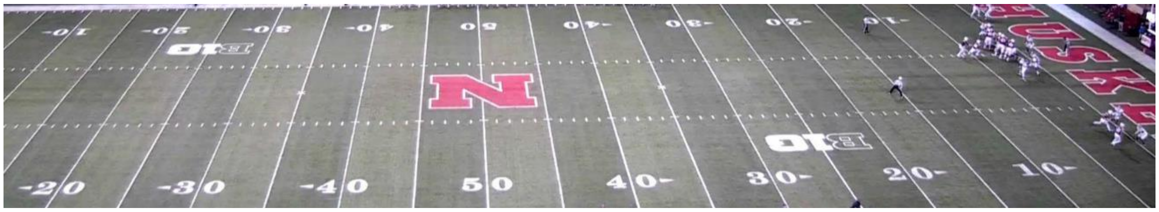

Figure 4.18: Snapshot of video misclassified as play. The video had equal amount of play and no play situation

We also ran 10-folds cross validation, with the entire 3000 video dataset. We present the cross-validated result Table 4.13, which has the average accuracy of 91.0%. Similar to the spectator space, we observe that the 10-folds cross validation has consistent classification efficacy through each iteration, showing that HoT is an effective feature to train classification model, independent of the training dataset.

Table 4.13 Cross validation result for spectator video classification, correct prediction shaded with green.:

Iteration #	**Play**			**No Play**			**Mixed**		
	Play	No Play	Mixed	Play	No Play	Mixed	Play	No Play	Mixed
1	100	4	2	7	91	1	4	5	86
2	92	4	2	2	96	3	3	6	92
3	103	5	2	6	88	3	6	9	78
4	88	5	8	3	87	5	4	12	88
5	83	2	4	1	96	1	6	8	99
6	85	5	3	2	98	3	4	2	98
7	96	2	2	11	88	6	2	5	88
8	90	6	2	2	89	12	3	4	92
9	92	3	5	6	90	7	4	5	88
10	95	6	4	3	88	5	8	4	87

4.8 Motion Trajectory Clustering Results

We implement trajectory clustering in both spectator and performer space in order to classify spatio-temporal clusters in both classes into different action categories. Before the training any classification model, we run preliminary experiments on both spectator and performer videos to determine the clustering parameters.

4.8.1 Parameter Selection

There were separate clustering experiments in spectator and performer space. For both spaces, we analyzed different values for clustering parameters before choosing parameters, which provided us with visually coherent clusters i.e. clusters having trajectory in spatially coherent areas, as well as highest number of clusters and lowest number of discarded trajectories. From Table 4.14, we observe that configuration with $\epsilon_{\perp}=10$, $\epsilon_{\parallel}=10$ and $N_{min}=10$, gives the highest number of clusters with least noise and which are visually coherent. So we chose these parameters for spectator space clustering.

Similarly, from Table 4.15, we choose that configuration with $€_{\perp}=10$, $€_{/\!/}=10$ and $N_{min}=5$ as it has highest number of clusters with least noise in performer space. We choose these parameters for the classification experiments with HoTC.

Table 4.14: Analysis of input parameters for DBSCAN-ST in spectator video

€⊥	**€∥**	**N_{min}**	**#Clusters**	**Clusters visually**	**% of Noise**
1	1	30	0	NA	100%
1	10	30	0	NA	100%
5	10	30	0	NA	100%
10	1	30	0	NA	100%
10	5	30	2	No	42%
10	10	30	3	Yes	35%
1	1	20	0	NA	100%
1	10	20	0	NA	100%
5	10	20	6	Yes	26%
10	1	20	0	NA	100%
10	5	20	3	No	40%
10	10	20	7	Yes	20%
1	1	10	0	NA	100%
1	10	10	0	NA	100%
5	10	10	8	No	25%
10	1	10	1	No	35%
10	5	10	6	No	28%
10	10	10	11	Yes	18%
1	1	5	0	NA	100%
1	10	5	0	NA	100%
5	10	5	9	Yes	26%
10	1	5	0	NA	100%
10	5	5	7	No	30%
10	10	5	9	Yes	28%

Table 4.15: Analysis of input parameters for DBSCAN-ST in performer video

€⊥	€‖	N_{min}	#Clusters	Clusters visually coherent	% of Noise
1	1	30	0	NA	100%
1	10	30	0	NA	100%
5	10	30	0	NA	100%
10	1	30	0	NA	100%
10	5	30	0	NA	100%
10	10	30	1	Yes	52%
1	1	20	0	NA	100%
1	10	20	0	NA	100%
5	10	20	0	NA	100%
10	1	20	0	NA	100%
10	5	20	0	NA	100%
10	10	20	2	Yes	45%
1	1	10	0	NA	100%
1	10	10	0	NA	100%
5	10	10	0	NA	100%
10	1	10	0	NA	100%
10	5	10	0	NA	100%
10	10	10	2	Yes	37%
1	1	5	0	NA	100%
1	10	5	0	NA	100%
5	10	5	1	Yes	45%
10	1	5	0	NA	100%
10	5	5	2	Yes	31%
10	10	5	2	Yes	25%

4.8.2 Clustering and Classification in Spectator Space

In spectator space, there were regions in the video where the spectators were excited and other regions where the spectators were passive and unexcited. We used DBSCAN-ST to implement clustering on the trajectories inside the video to find and classify those regions as active, passive or mixed. An illustration of clusters in spectator video is presented in Figure 4.19.

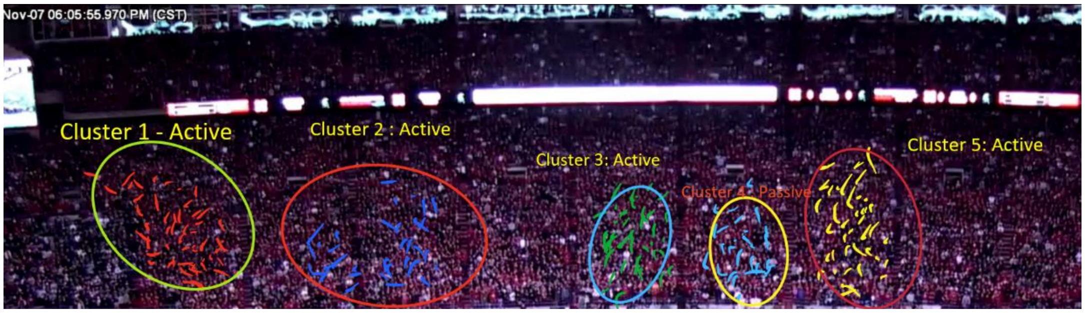

Figure 4.19: Clusters in spectator video of active class

Table 4.16: Clustering result on active, passive and mixed spectator videos

Video Category	# Videos	# Clusters	# Active Cluster/# verified		# Passive Cluster/ # verified		# Mixed Cluster/ # verified	
Active	35	326	305	290	15	12	6	4
Passive	35	255	17	13	228	218	10	8
Mixed	30	248	89	80	99	93	60	50

We can see from Table 4.16 that there are different regions in single video that are active, passive or mixed. Although, the majority of clusters in video are of same class for active and passive video, we observe that mixed cluster has equal number of active, passive and mixed regions.

We can verify that the cluster classification improves accuracy over video classification, as the overall accuracy of prediction on the clusters increased by 87.5% to 92.5%. This is a significant improvement, and is helped by the fact that clusters provide a finer representation of activity in the spectator space as spectator space are very large and diverse.

4.8.3 Clustering and Classification in Performer Space

In performer space, there were regions in the video were the groups of players were concentrated and regions where there were lone players and official, who were not involved in the action. We clustered the performer videos to find regions where players in order to optimize our classification, and give a more fine-grained classification of our dataset. We expect that classifying clusters instead of the entire video is more accurate as DBSCAN-ST finds region in the videos that are significant as well as remove region with noise. We see some example of clustering in performer space in Figure 4.20 and Figure 4.21.

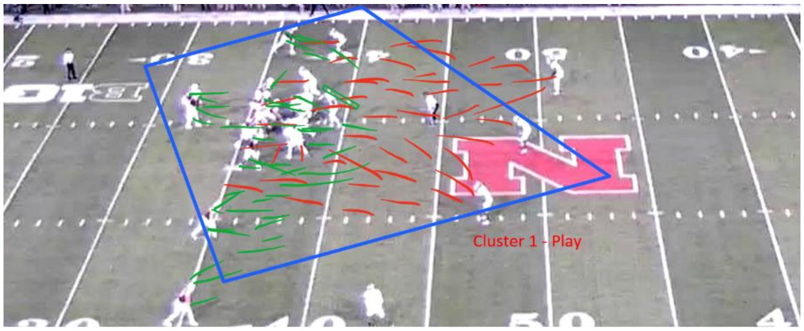

Figure 4.20: Trajectory clustering in performer video of play class

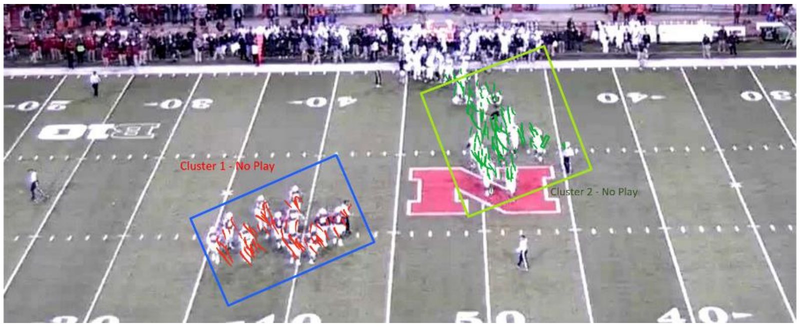

Figure 4.21: Trajectory clustering in performer video of no play class

We used the same process used in spectator space to build and test the classifier for performer clusters. The result from those experiments are presented in Table 4.17.

Table 4.17: Clustering result on play, no play and mixed performer videos

Video Category	# Videos	# Clusters	# Play Cluster/ # verified		# No Play Cluster/ # verified		# Mixed Cluster/ # verified	
Play	35	48	42	40	5	2	1	1
No Play	35	58	3	2	52	45	3	1
Mixed	30	54	18	15	21	18	25	22

The overall accuracy of prediction for performer videos was 91.25%. There are several implications of the results in Table 4.17. First, by comparing the number of clusters in Table 4.16 we see that there are fewer numbers of clusters in performer space. This is because, for our dataset, the movement players either start or end in same region. This allows most trajectories to be in the same cluster by the DBSCAN-ST clustering. Second, our clustering algorithm removes noise or unrelated trajectories from the dataset, which we can be observed from the marginal improvement in the accuracy of clusters classification.

4.9 Classification with Deep Learning

We use deep 3-dimensional convolutional networks (C3D) developed by Tran el al [26] to train a video classifier for both spectator and performer videos. In Section 4.9.1 and 4.9.2, we describe C3D architecture and classification experiments on spectator and performer videos, respectively. Finally, in Section 4.9.3, we present the comparison of classification efficacy between C3D, HoT and HoTC and their runtime.

4.9.1 C3D Architecture

C3D is a holistic classification technique that learns spatio-temporal features from videos to build a linear classifier. Its composed of the following types of layers.

- *Convolutional:* This is the most computational layer in the deep learning architecture. It consists of a set of spatially small learnable filters which convolve through the entire space of video frame, as well through all the frames in the video. In C3D, it produces a 3-dimensional activation map, as it is a 3-dimensional convolution layer.
- *Pooling:* This is a dimension-reducing layer, which is inserted between convolutional layers to reduce the outputs from convolutional layer by different polling strategy like max-pooling, average-pooling or norm-pooing.
- *Fully Connected:* This layer is exactly same as any normal neural network layer as it contains connection to all output from the previous layers.

It is a deep learning network with 16 different layers of which eight are *convolutional* layers, five are *max-pooling* layers and two *fully connected* layers. The output of this deep network is a 4096-dimension video descriptor which is then used by a SVM to make a prediction. The network diagram for C3D feature generation is given in Figure 4.22 below. This feature can then be used to train SVMs to get the classification of a video. SVM is also included inside the C3D architecture, making this a holistic process, with video as an input and a class as an output.

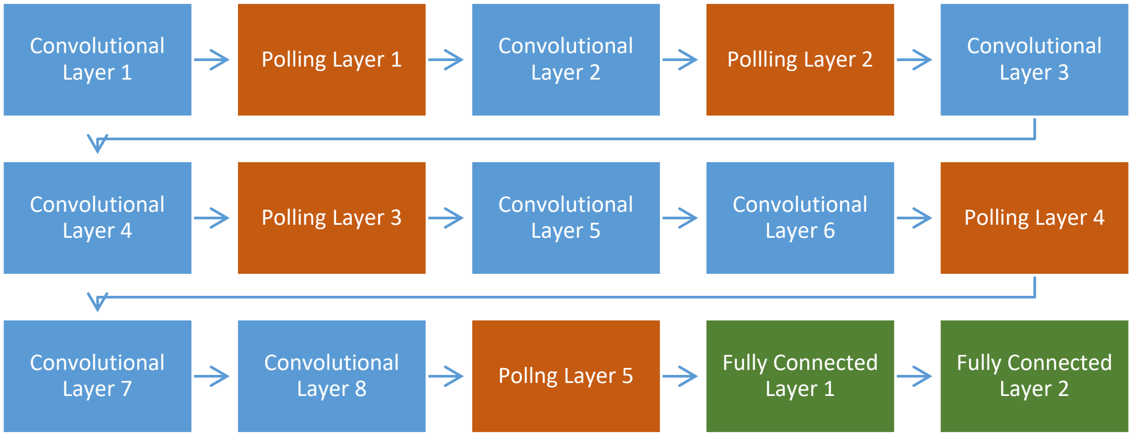

Figure 4.22: C3D network architecture

4.9.2 Classification Results

We trained a C3D classifier with 1500 spectator videos, and tested the classifier with another 1500 videos. The overall performance of C3D was 67.13% accurate with roughly uniform incorrect prediction for all 3 classes. Table 4.18 is the confusion matrix for this experiment. We see that the class precision is highest for active class but it also had the least true positive rate i.e. most active videos were classified as active, and high number of other videos were also classified as active.

Table 4.18 Confusion matrix – C3D classification for spectator video

	Active	**Passive**	**Mixed**	**Class Precision**
Active	340	88	72	68.00%
Passive	96	324	80	64.80%
Mixed	92	65	343	64.60%
Class Recall	64.39%	67.92%	69.29%	

Similarly, we train another C3D classification model with 1500 performer video, 500 from each of play, no play and mixed category. The overall classification accuracy was 69.93%. In table 4.19, we have the confusion matrix for this experiment. We see that positive classification for all three categories were within 8 percentage points from one another with highest class precision for mixed category. The recall value (true positive rate) was similar for no play and mixed videos whereas lower for play videos. C3D classified higher number of other videos erroneously as play video.

Table 4.19: Confusion matrix – C3D classification for performer video

	Play	**No Play**	**Mixed**	**Class**
Play	348	77	75	69.60%
No Play	99	326	75	65.20%
Mixed	82	53	365	73.00%
Class Recall	65.78%	71.49%	70.8	

4.9.3 Efficacy and Runtime Comparison with HoT and HoTC

We compared the efficacy of deep learning based C3D classifier [26] with both our classification approaches, i.e. HoT and HoTC using a NaïveBayes classifier. We observed that both HoT and HoTC were more accurate in their predictions. From Table 4.20, we see that HoT and HoTC are consistently better in classification of both spectator and performer videos by 18-22 %. Similarly, from Table 4.21 and Table 4.22, we see that class wise accuracy of HoT and HoTC are better than C3D for both spectator and performer space.

Table 4.20: Classification accuracy between Deep Learning, HoT and HoTC on SPS videos

Space	**Deep Learning - C3D**	**HoT**	**HoTC**
Spectator	67.13%	87.5%	92.5%
Performer	69.93%	91.0%	91.25%

Table 4.21: Confusion matrix for classification in spectator space using HoT, HoTC and Deep Learning (C3D)

	Active	Passive	Mixed	Class Accuracy
Active	426 383 340	32 12 88	42 3 72	85.6% 98.7% 64.6%
Passive	51 19 96	433 313 324	16 1 80	86.6% 93.9% 64.8%
Mixed	60 9 92	21 17 65	419 62 343	83.2% 70.4% 64.6%
Class Recall	79.9% 93.1% 64.3%	89.1% 91.5% 67.9%	87.8% 93.9% 69.2%	

Table 4.22: Confusion matrix for classification in performer space using HoT, HoTC and Deep Learning (C3D)

	Active	Passive	Mixed	Class Accuracy
Active	448 57 348	14 13 77	38 2 75	89.6% 91.9% 69.6%
Passive	30 4 99	458 55 326	12 3 75	91.6% 88.7% 65.2%
Mixed	40 2 82	25 10 53	435 24 365	87.0% 66.6% 73.0%
Class Recall	84.8% 90.4% 65.7%	92.1% 75.3% 71.4%	89.6% 82.7% 70.8%	

Even though the performance of our feature-based classification was better than C3D, we do not have enough data to understand this uncharacteristic performance of C3D. This is due to two reasons – first, lack of enough training and testing data for deep learning based experimentations and second, prohibitively slow convergence of C3D on our dataset. We leave the exploration of these results as future work.

We compare C3D [20] which is a deep learning based method with our approach of video classification in terms of efficacy and performance. Since deep learning based methods have highest classification efficacy in video classification on standard datasets like Hollywood and UCF-Sports, we chose deep learning based methods to make these comparisons.

The runtime requirements of deep learning based methods are well document in literature [26, 27]. Since deep learning based classifiers require comparatively high amount of runtime, we also compare the time required to train all models (deep learning, HoT and HOTC NaïveBayes) in our experimental setup. We observed that HoT and HoTC required between 8-13 times less C3D. We see in Table 4.21 that the runtime of HoTC is longer than HoT as HoTC requires DBSCAN-ST clustering. This was expected as the runtime complexity of our algorithms were at most 2nd degree polynomial whereas the runtime complexity of C3D is exponential on the number of layers present in C3D [26].

Table 4.23 Experiment runtime between Deep Learning, HoT and HoTC on SPS videos

Space	Deep Learning C3D (hours)	HoT (hours)	HoTC (hours)
Spectator	348	32	39
Performer	336	24	25

4.10 Discussion of Result

Based on the observations from Section 4.3 and Section 4.4, we establish the properties of spectator and performer space with respect to their motion trajectories, as well as how significantly they differ from one another. We explored the motion trajectory features in both spaces and observed that spectator space has more motion trajectory because of two reasons: (a) it contains more people, and (b) high density of people causes trajectory fragmentation.

We also studied the properties of several action categories of spectator and performer videos and conclude that those action categories induce distinctly different motion trajectories. We see that the active videos have more vertical trajectories whereas passive videos have horizontal trajectories in spectator space. Similarly, in performer space, we find that play videos have more horizontal and longer trajectories than passive video.

The results from Section 4.6, 4.7 and 4.8 show that HoT and HoTC based features are highly effective in classifying performer and spectator videos into different action categories. We see that spectator videos contain multiple classes of activities within the same video. So, it is effective to use density based clustering and group related trajectories together and classify the clusters instead of the entire video, to prevent from different region in the video with different classes of activities being classified into same class.

Also, for performer videos, HoTC does not make as significant improvement in classification. This is because most performer videos contain only one cluster per video as most performers perform actions together and have very low probability of doing entirely different things at the same time.

Finally, from Section 4.9, we see that the classification efficacy of HoT and HoTC based classifier outperforms even state of art deep learning classification models (C3D). Also, the runtime of our classification techniques are significantly lower than that of C3D.

Chapter 5

Summary and Future Work

5.1 Summary

In this thesis, we defined a new class of videos, called Spectator Performer Space (SPS) and analyzed its properties in terms of density, size, behavior and complexity. We have developed approaches to classify the videos based on the activities of the performers and spectators. We proposed a set of novel features based on individual motion trajectories as well as trajectory clusters that are used for classification. We have extended a well-known density based clustering algorithm suitable for clustering short trajectories, common in SPS domain.

The algorithms were evaluated using a large dataset of sports videos. The results show trajectory length and orientation are very effective in accurately characterizing both spectator and performer videos. The properties of trajectory clusters were also effective in classifying the videos based on their activities.

5.2 Direction of Future Research

We have defined a new class of videos in this research and provided solutions to some fundamental problems. There are several avenues for extending this work along several different directions.

Extending trajectories from short, straight lines to more complex curves will enable the representation of complex trajectories more accurately. Similarly, using a continuous representation for orientation of trajectories, instead of a discrete one would lead to greater accuracy. Additionally, we can extend the trajectory extraction and characterization process to include videos from multiple angles. This would require us to be able to identify same trajectories in different videos but provide a more comprehensive representation of motion trajectory in the video.

Velocity and acceleration are also important feature to characterize the motion of objects in a scene. These two features can help identify the rate of change in crowd behavior i.e. how quickly are the spectators changing from active to passive and vice versa. Our feature set of trajectory length and orientation can include velocity and acceleration to make it more complete.

Similarly, we can extend this work to implement processes to automatically segment video segments, both temporally and spatially, into different action regions. As larger spaces can have multiple action over time and space, automatic segmentation of videos along time and space is highly desirable.

Additionally, we have only focused on the emotion of the crowd at a coarse resolution. Analyzing the crowd emotions in greater details and classifying them would also be beneficial in many applications. Identification of outliers in both the spectator and performer spaces would also be useful.

Also, a more comprehensive evaluation of the approaches with a larger number of videos from diverse domains would be a useful exercise. Larger collection of videos will also be critical to improving the performance of the deep-learning approach. Segmentation of the videos accurately into activity based episodes will also be helpful in this context.

Appendix A

A.1 Analysis of trajectory distribution between different classes of spectator videos.

Table A.1.1: Variance analysis on active vs passive vs mixed spectator videos trajectory distribution with respect to length. #active, #passive and #mixed are number of trajectories in active, passive and mixed video respectively of length corresponding to the length parameter given in their respective rows.

Length	Alternative Hypothesis	Significance
Long	#active > #passive	Yes
Medium	#active > #passive	Yes
Short	#passive > #active	Yes
Long	#active > #mixed	Yes
Medium	#active > #mixed	Yes
Short	#mixed > #active	Yes
Long	#mixed > #passive	Not Statistically Significant
Medium	#mixed > #passive	Yes
Short	#mixed > #passive	Yes

Table A.1.2: Variance analysis on active vs passive spectator videos trajectory distribution with respect to orientation. #active and #passive are number of trajectories in active and passive video respectively of orientation corresponding to the value given in their respective rows.

Orientation	Alternative Hypothesis	Significance
0-20	#active > #passive	Yes
20-40	#active > #passive	Yes
40-60	#active > #passive	Yes
60-80	#active > #passive	Yes
80-100	#active > #passive	Not Statistically Significant
100-120	#active > #passive	Not Statistically Significant
120-140	#active > #passive	Yes
140-160	#active > #passive	Yes
160-180	#active > #passive	Yes

Table A.1.3: Variance analysis on active vs passive spectator videos trajectory distribution with respect to orientation. #mixed and #passive are number of trajectories in mixed and passive video respectively of orientation corresponding to the value given in their respective rows.

Orientation	Alternative Hypothesis	Significance
0-20	#mixed > #passive	Yes
20-40	# mixed > #passive	Yes
40-60	# mixed > #passive	Yes
60-80	# mixed > #passive	Not Statistically Significant
80-100	# mixed > #passive	Not Statistically Significant
100-120	# mixed > #passive	Not Statistically Significant
120-140	# mixed > #passive	Not Statistically Significant
140-160	# mixed > #passive	Yes
160-180	# mixed > #passive	Yes

Table A.1.4: Result of variance analysis on active vs passive spectator videos trajectory distribution with respect to orientation. #active and #mixed are number of trajectories in active and mixed video respectively of orientation corresponding to the value given in their respective rows.

Orientation	Alternative Hypothesis	Significance
0-20	#active > # mixed	Yes
20-40	#mixed > # active	Yes
40-60	#active > # mixed	Yes
60-80	#active > # mixed	Not Statistically Significant
80-100	#mixed > # active	Yes
100-120	#active > # mixed	Yes
120-140	#active > # mixed	Not Statistically Significant
140-160	#mixed > # active	Yes
160-180	#active > # mixed	Yes

A.2 Analysis of trajectory distribution between different classes of performer videos.

Table A.2.1: Result of variance analysis on play vs no play vs mixed performer videos trajectory distribution with respect to length. #play, #no play and #mixed are number of trajectories in play, no play and mixed video respectively of length corresponding to the length parameter given in their respective rows

Length	Alternative Hypothesis	Significance
Long	#play > #no play	Yes
Medium	#play > #no play	Yes
Short	#no play > #play	Yes
Long	#mixed > #no-play	Not Statistically Significant
Medium	#mixed > #no-play	Yes
Short	#mixed > #no-play	Yes
Long	#play > #mixed	Yes
Medium	#play > #mixed	Yes
Short	#mixed > #play	Yes

Table A.2.2: Result of variance analysis on play vs no play performer videos trajectory distribution with respect to orientation. #play and #no play are number of trajectories in play and no play video respectively of orientation corresponding to the value given in their respective rows.

Orientation	Alternative Hypothesis	Significance
0-20	#play > #no play	Yes
20-40	#play > #no play	Yes
40-60	#play > #no play	Yes
60-80	#play > #no play	Yes
80-100	#no play > #play	Yes
100-120	#play > #no play	Not Statistically Significant
120-140	#play > #no play	Yes
140-160	#play > #no play	Yes
160-180	#play > #no play	Yes

Table A.2.3: Result of variance analysis on mixed vs no play performer videos trajectory distribution with respect to orientation. #mixed and #no play are number of trajectories in mixed and no play video respectively of orientation corresponding to the value given in their respective rows.

Orientation	Alternative Hypothesis	Significance
0-20	#mixed > #no play	Yes
20-40	#mixed > #no play	Yes
40-60	#mixed > #no play	Yes
60-80	#mixed > #no play	Yes
80-100	#no play > #mixed	Yes
100-120	#mixed > #no play	Yes
120-140	#mixed > #no play	Not Statistically Significant
140-160	#mixed > #no play	Yes
160-180	#mixed > #no play	Yes

Table A.2.4: Result of variance analysis on play vs mixed performer videos trajectory distribution with respect to orientation. #play and #mixed are number of trajectories in play and mixed video respectively of orientation corresponding to the value given in their respective rows.

Orientation	Alternative Hypothesis	Significance
0-20	#play > #mixed	Yes
20-40	#play > #mixed	Yes
40-60	#play > #mixed	Yes
60-80	#play > #mixed	Yes
80-100	#play > #mixed	Not Statistically Significant
100-120	#play > #mixed	Not Statistically Significant
120-140	#play > #mixed	Yes
140-160	#play > #mixed	Yes
160-180	#play > #mixed	Yes

Bibliography

(1) Zhan B, Monekosso DN, Remagnino P, Velastin SA, Xu L. Crowd analysis: a survey. Mach Vision Appl 2008;19(5-6):345-357.

(2) Solmaz B, Moore BE, Shah M. Identifying behaviors in crowd scenes using stability analysis for dynamical systems. Pattern Analysis and Machine Intelligence, IEEE Transactions on 2012;34(10):2064-2070.

(3) Helbing D, Molnar P. Social force model for pedestrian dynamics. Physical review E 1995;51(5):4282.

(4) Mondal P. Crowd: Meaning, Process of Formation and Classification. 2015; Available at: http://www.yourarticlelibrary.com/essay/crowd-meaning-process-of-formation-and-classification/31290/. Accessed 05/18, 2016.

(5) Fradi H, Dugelay J. Towards crowd density-aware video surveillance applications. Information Fusion 2015;24:3-15.

(6) Ali S, Shah M. Floor fields for tracking in high density crowd scenes. Computer Vision–ECCV 2008: Springer; 2008. p. 1-14.

(7) Still GK. Crowd Safety and Risk Analysis. 2016; Available at: http://www.gkstill.com/ExpertWitness/CrowdDisasters.html. Accessed 05/15, 2016.

(8) Trajectory clustering: a partition-and-group framework. Proceedings of the 2007 ACM SIGMOD international conference on Management of data: ACM; 2007.

(9) O'Hara S, Draper BA. Introduction to the bag of features paradigm for image classification and retrieval. arXiv preprint arXiv:1101.3354 2011.

(10) Kotsiantis SB, Zaharakis I, Pintelas P. Supervised machine learning: A review of classification techniques 2007.

(11) Crowd behavior recognition for video surveillance. Advanced Concepts for Intelligent Vision Systems: Springer; 2008.

(12) Density-aware person detection and tracking in crowds. Computer Vision (ICCV), 2011 IEEE International Conference on: IEEE; 2011.

(13) Polus A, Schofer JL, Ushpiz A. Pedestrian flow and level of service. J Transp Eng 1983;109(1):46-56.

(14) Histograms of oriented gradients for human detection. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on: IEEE; 2005.

(15) Wang H, Kläser A, Schmid C, Liu C. Dense trajectories and motion boundary descriptors for action recognition. International journal of computer vision 2013;103(1):60-79.

(16) An HOG-LBP human detector with partial occlusion handling. Computer Vision, 2009 IEEE 12th International Conference on: IEEE; 2009.

(17) Baeza-Yates R, Ribeiro-Neto B. Modern information retrieval. : ACM press New York; 1999.

(18) Nicholson CE, Roebuck B. The investigation of the Hillsborough disaster by the Health and Safety Executive. Saf Sci 1995;18(4):249-259.

(19) An iterative image registration technique with an application to stereo vision. IJCAI; 1981.

(20) Large-scale video classification with convolutional neural networks. Proceedings of the IEEE conference on Computer Vision and Pattern Recognition; 2014.

(21) Bengio Y. Learning deep architectures for AI. Foundations and trends® in Machine Learning 2009;2(1):1-127.

(22) Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems; 2012.

(23) Peekaboom: a game for locating objects in images. Proceedings of the SIGCHI conference on Human Factors in computing systems: ACM; 2006.

(24) LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE 1998;86(11):2278-2324.

(25) Krizhevsky A, Hinton G. Convolutional deep belief networks on cifar-10. Unpublished manuscript 2010;40.

(26) Tran D, Bourdev L, Fergus R, Torresani L, Paluri M. Learning spatiotemporal features with 3d convolutional networks. arXiv preprint arXiv:1412.0767 2014.

(27) Soomro K, Zamir AR, Shah M. UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 2012.

(28) Piciarelli C, Foresti GL. On-line trajectory clustering for anomalous events detection. Pattern Recog Lett 2006;27(15):1835-1842.

(29) A coarse-to-fine strategy for vehicle motion trajectory clustering. Pattern Recognition, 2006. ICPR 2006. 18th International Conference on: IEEE; 2006.

(30) Antonini G, Thiran JP. Counting pedestrians in video sequences using trajectory clustering. Circuits and Systems for Video Technology, IEEE Transactions on 2006;16(8):1008-1020.

(31) Trajectory clustering for motion prediction. Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on: IEEE; 2012.

(32) Tra-dbscan: a algorithm of clustering trajectories. Applied Mechanics and Materials: Trans Tech Publ; 2012.

(33) Laptev I. On space-time interest points. International Journal of Computer Vision 2005;64(2-3):107-123.

(34) A combined corner and edge detector. Alvey vision conference: Citeseer; 1988.

(35) Visual attention detection in video sequences using spatiotemporal cues. Proceedings of the 14th annual ACM international conference on Multimedia: ACM; 2006.

(36) Willems G, Tuytelaars T, Van Gool L. An efficient dense and scale-invariant spatio-temporal interest point detector. Computer Vision–ECCV 2008: Springer; 2008. p. 650-663.

(37) Recognizing realistic actions from videos "in the wild". Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on: IEEE; 2009.

(38) HMM based classification of sports videos using color feature. Intelligent Systems (IS), 2012 6th IEEE International Conference: IEEE; 2012.

(39) Rabiner LR. A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE 1989;77(2):257-286.

(40) Baker S, Matthews I. Lucas-kande 20 years on: A unifying framework: Part 1 2002.

(41) Human detection using oriented histograms of flow and appearance. European conference on computer vision: Springer; 2006.

(42) A density-based algorithm for discovering clusters in large spatial databases with noise. Kdd; 1996.

(43) Chen J, Leung MK, Gao Y. Noisy logo recognition using line segment Hausdorff distance. Pattern Recognit 2003;36(4):943-955.

(44) Fradkin D, Muchnik I. Support vector machines for classification. Discrete methods in epidemiology 2006;70:13-20.

(45) Awad M, Motai Y. Dynamic classification for video stream using support vector machine. Applied Soft Computing 2008;8(4):1314-1325.

(46) A study of cross-validation and bootstrap for accuracy estimation and model selection. Ijcai; 1995.

(47) Rokach L. Ensemble-based classifiers. Artif Intell Rev 2010;33(1-2):1-39.

(48) Schmidhuber J. Deep learning in neural networks: An overview. Neural Networks 2015;61:85-117.

(49) Caffe: Convolutional architecture for fast feature embedding. Proceedings of the 22nd ACM international conference on Multimedia: ACM; 2014.