ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL SCIENCES

Real-time Shoplifting Detection from Surveillance Video

Daniel Sahle Chemere

A Thesis Submitted to the Department of Computer Science in Partial Fulfillment for the Degree of Masters of Science in Computer Science

05 October, 2018

ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL SCIENCES

Daniel Sahle Chemere

Advisor: Yaregal Assabie (PhD)

This is to certify that the thesis prepared by Daniel Sahle Chemere, titled: *Real-time Shoplifting Detection from Surveillance Video* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the university and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

Name	Signature
Advisor: Yaregal Assabie (PhD)	_____
Examiner: _____	_____
Examiner: _____	_____

05 October, 2018

Abstract

With surveillance cameras being ubiquitous in most big stores while shoplifting breaks owners' banks, better prevention mechanisms for the crime is vital more than ever. This can be achieved through an efficient automatic detection surveillance system that can detect this event.

The current methods employed in the industry are not efficient as human operators scanning a lot of screens have their own shortcomings. Human labor is also getting more expensive, especially in urban places where such stores exist in abundance. Existing methods of activity detection do not address the problem as each action has unique characteristics and intricate details that it has to be modeled independently.

This thesis introduces a novel hybrid based real-time shoplifting detection architecture that detects the event of shoplifting from surveillance videos. The model, using CNN classification and optical flow features from the sequence of frames, makes use of different features of the event that is learned from video examples and applies different techniques to this to detect when the event occurs in sight of the cameras. Moreover, a joint-based rule-based method of detection of joint proximity is designed.

To show the effectiveness of the system, a prototype is developed and tested with a dataset we prepared for the purpose of this specific thesis. The analysis of the evaluation indicates that the system provides an efficient automatic real-time shoplifting detection with 55% recall and 60% precision.

Keywords: Event detection; Shoplifting; Shoplifting detection; Joint-based Optical flow; Event Fusion; Event Modelling

Dedication

Verily, verily, I say unto you, except a grain of wheat fall into the ground and die, it abideth alone: but if it die, it bringeth forth much fruit.

John 12:24

Acknowledgment

Foremost, I would like to express my heartfelt gratitude to my advisor Dr. Yaregal Assabie, whose constant pressure, his patience, motivation and guidance made this paper possible. It is only with his constant assistance and unremitting advice that this work transformed from a fantasy and a figment of my imagination to a reality that can be wholly grasped.

I would like to offer my special thanks to all my friends who have supported and helped me in the thesis with their ingenious ideas and assistance. I am also very grateful to all my friends and family who have kept me busy with other life matters without which this thesis would have been completed much sooner.

I am particularly grateful for the assistance given by Mr. Berhanu Abebe, Mr. Meareg Abreha, Mr. Berhanyihun Amanuel and Mr. Kibrewossen Yitbarek who had relentlessly listened to my ideas and supported me a lot in developing them. I am indebted to Mr. Amanuel Getaneh, Mr. Yonas Tadesse and the personnel at Titi Supermarket who have contributed their priceless help during the dataset preparation. My special thanks are extended to the staff and students of the Computer Science department at Addis Ababa University. Finally, I want to give regards to all the people that have knowingly or unknowingly helped me shape the thesis into its current form.

Table of Contents

List of Tables

List of Algorithms

List of Figures

Acronyms and Abbreviations

CCTV	Closed-circuit Television
CCV	Color Coherence Vector
CIF	Common Intermediate Format
CM	Color Moments
CNN	Convolutional Neural Networks
CoHOGs	Co-occurrence Histograms of Oriented Gradients
ConvNet	Convolutional Networks
CRF	Conditional Random Fields
DBN	Dynamic Bayesian Network
DSLR	Digital Single-Lens Reflex
EAS	Electronic Article Surveillance
EM	Expectation Maximization
HD	High Definition
HSV	Hue, Saturation, Value
JRoG	Joint Ranking of Granules
LMDB	Lightning Memory-Mapped Database
LPCRF	Latent Pose Conditional Random Field
MAP	Maximum A posteriori Probability
PF	Particle Filter
PTZ	Pan-Tilt-Zoom Camera
QCIF	Quarter Common Intermediate Format
RFID	Radio Frequency Identification
RGB	Red, Green, Blue
SCD	Scalable Color Descriptor
SED	Surveillance Event Detection
SIFT	Scale Invariant Feature Transform
SVM	Support Vector Machines
TRECVid	Text Retrieval Conference Video Retrieval Evaluation

Chapter 1: Introduction

1.1 Overview

The need for security and the wide range of affordable technologies have enabled us to monitor and store a massive amount of video data. With the advancement and availability of video recording devices, video data has increased sharply in volume. The constant threats from terrorists and different type of crimes on the rise, using surveillance cameras cannot be undermined. However, these data that we record are enormous in volume and analyzing them by manually watching them from the beginning to the end will require a lot of time and labor.

Although surveillance cameras are already ubiquitous in banks, stores, and parking lots, video data currently is usually used only after a crime as a forensic tool, thus losing its primary benefit as an active, real-time medium [1]. As one of the many crimes that are frequent in stores, shoplifting gets a good attention. Shoplifting is the act of knowingly obtaining goods from an establishment in which they are displayed for sale, without paying for them. Shoplifting is a big problem for store owners and many other parties including the government, the police and court. According to a research by National Association for Shoplifting Prevention, 1 in 11 people shoplift. Moreover, it is reported that shoplifters are caught only once in every 48 times they steal [2].

The National Retail Security Survey conducted by the University of Florida [3] has reported that shoplifting cost 10 Billion USD in 2002 in the United States alone. To alleviate this problem of gigantic proportions, there are a lot of preventive mechanisms employed by store owners. The same report listed the most common prevention systems used to fight against shoplifting. From the total retail stores surveyed, around three fourth of them used visible live CCTV cameras, half used hidden CCTV, half used observation mirrors and around forty percent used Electronic Article Surveillance (EAS) tags.

Although the use of these loss preventive systems decreases the number of shoplifting cases, it is still one of the biggest losses for retail stores worldwide. Even if the use of surveillance systems is not new to the field of shoplifting detection, it is usually used to manually review the recorded tapes after the feat had already happened and after the criminal had runaway.

These systems might be used in identifying frequent shoplifting offenders and might help the owners in identifying areas in the store which are targeted frequently by the shoplifters. Nonetheless, these systems do not provide any mechanisms for preventing an ongoing shoplifting.

In the presence of CCTV, stores just use them as an evidence to analyze only after finding out a crime has been committed. At best, detection of events from surveillance video is usually done manually where officers and security staff watch live feeds and footages of an area that needs security. Even though this has been and still is true for many scenarios, it has its own shortcomings. The first drawback that comes to mind is that personnel, being humans, can be bored and/or tired and at times miss some moments which could be very important ones that contain a lot of information. The second and biggest challenge is that now the technology is getting more affordable, most times, cheaper than that of a human employee.

An additional example where manual processing would prove to be ineffective is where a single human operator operates so many TV screens simultaneously and obviously could not observe everything. Therefore, detection of different events like shoplifting from surveillance cameras are better done automatically instead of manually for better accuracy and economic reasons.

The automatic analysis and detection of specific events in videos is important for understanding the semantic content of videos. Generally speaking, event detection involves identifying and extracting different actions from a video and then trying to extract meaning and semantics from those actions. Video event detection allows intelligent indexing of video content based on events that are extracted from the video [3].

As a difficult problem to tackle, many literature have been dedicated to video event detection and surveillance event detection topics. Different methods were employed on it from different researchers. A typical event detection system follows many or all of the following set of steps to reach its goal; Extracting of frames from the video, Preprocessing (noise removal, resize), Object segmentation, Feature Extraction and Activity detection. Depending on the application of the system, human tracking can be an additional task.

There have been numerous research efforts reported for various applications based on human activity recognition, more specifically, home abnormal activity [4], ballet activity [5], tennis

activity [6], [7], soccer activity [8], human gestures [10], sport activity [9][12], human interaction [13], pedestrian traffic [10] and other topics.

Another set of literature initiated and evaluated by TRECVID surveillance event detection and using the data provided by the project have been done on some specific tasks. Besides those, there are no literature directly related to the actions and events of shoplifting to the best of the researcher's knowledge.

Though a lot of work has been done in the field of event detection from video, it is significantly not enough as compared to the importance and difficulty of the area and is still a hot research topic. The main challenges with automatic event detection are the quality of the video, presence of occlusion as well as the amount of noise present. Additionally, translating low-level input into a semantically meaningful event description is a tough, sometimes even tricky, task [11].

1.2 Statement of the Problem

Shoplifting, as a major problem in everyday life, should be detected for the betterment of business as a whole. The puzzle needs better results from preventive systems to detect and resolve issues of shoplifting which can be done by automatically analyzing videos from surveillance in real-time as it happens.

Previous works on event detection have shown significance improvements in activity recognition for specified action. Good examples for these are works initiated and evaluated by TRECVID surveillance event detection (SED) task [12], which has seven types of human activities namely CellToEar, Embrace, ObjectPut, PeopleMeet, PeopleSplitUp, PersonRuns, and Pointing. These works are trained specifically to recognize these events and their respective designs are also tailored to identify such events that they are developed to resolve.

To the researcher's knowledge, there is no work so far done that directly addresses the problem by detecting shoplifting automatically from videos. Hence, we attempt to provide an architecture for the automatic detection of the event from surveillance video. Subsequently, the works already present for other tasks could not be used for these purposes without entire modifications of a number of components that alter the whole integrity of their respective works. Also, due to the complexity of the gestures and hand movements that represent an

action of shoplifting, the works could not be directly applied for actions of shoplifting because some are too general and many are designed for specific action not directly related to this event. This research proposes an automatic detection of shoplifting from surveillance cameras.

1.3 Objective

General Objective

The general objective of this research is to provide an architecture for the development of a Real-Time Shoplifting Detection System from Surveillance Video.

Specific Objectives

The specific objectives devised to achieve the general objective are:

- review related literature in the area of image processing and video analysis as well as event detection
- collect video data from surveillance
- study about properties and characteristics of gesture and hand movements specific to shoplifting
- identify components of video event detection systems and design Shoplifting Detection System
- adopt some existing tools, techniques and approaches in preprocessing, object segmentation, feature extraction and representation.
- develop a prototype that shows the viability of the proposal
- evaluate the performance of the shoplifting detector

1.4 Methodology

The main goal of this thesis work is to design an automatic real-time shoplifting detection model using feeds from surveillance cameras and implement part of it. In order to accomplish the specified objectives of the study, both general and specific, different methods will be employed.

Literature Review

An extensive review of relevant literature will be conducted to acquire a deeper understanding of the research area and its problem domains. We will use systematic literature review methods to identify the works that are related to our topic and previous researches in the areas of event detection, automatic surveillance systems and other related issues will be investigated to visualize their importance towards this research. Existing publications related to this research work will also be assessed to identify and point direction in providing solution to identified problems.

Data Collection

Video data from surveillance cameras that contain different actions of shoplifting will be collected from different stores. Alternatively, new videos of such endeavors could be recorded using a camera.

Design

Using the inputs from conducted review of literature as well as shoplifting and other gestures from the surveillance data, useful models and design requirements will be identified to arrive at sensible solutions. Based on these requirements, a model of context and design of architecture will be proposed which can guide implementation of certain applications or a prototype.

Tools

Tools and techniques that can fulfill the design of the architecture shall then be labeled for the implementation of the prototype system. Python and OpenCV library will be used in the development process and Caffe framework for deep learning. Additionally, existing techniques and approaches will be adopted in the different stages of the process.

Prototype Development

The prototype should consider the necessary identified services to prove the significance of the proposed solution so that it should carefully follow the specification that will be provided

in the model and architecture. To validate this prototype and the significance of the current work, testing and evaluation techniques shall be identified and used.

1.5 Scope and Limitation

Detection of actions from a video feed contains a lot of tasks that range from simple to very complex. This research will not focus on multiple camera feeds as it only attempts to detect events from a single camera. Additionally, the work does not consider movement of the camera and only takes into account a fixed motionless camera at work. Moreover, when an action is occluded by any object that obscures clear sight of the endeavor, the system is not expected to neither identify nor understand it.

The architecture we are proposing only attempts to identify shoplifting as pocketing on an item; it does not address consumption of the item in the store or when a shoplifter puts on an item with the intent of stealing.

1.6 Application of Results

After a successful development of a system, supermarkets, department stores, convenient stores as well as other retail stores can benefit from it as it can help detect and prevent shoplifting. Such a system helps locate and identify in real-time when such an endeavor is committed and is expected to notify employees of the deed. Hence, the system has the potential to prevent or at least to reduce the amount of shoplifting incidents in retail stores which has become a big headache for customers, store owners and other bodies.

1.7 Thesis Organization

The remainder of this thesis is organized in the following manner. In Chapter 2, the Literature Review section, detailed review of the core theories that have significance to our thesis are presented. Chapter 3, the Related Work, zooms in works that are related to ours in that they try to tackle similar problems of event detection and these works are reviewed and critically analyzed to show their importance and their gaps. Chapter 4, the Design, stages our major body of work and explains our proposal in adequate depth; algorithms, frameworks and architecture are shown and explanations and descriptions are offered. Results of the different tests we get from the system is reported in Chapter 5, the Experiment, in which different

metrics are used to evaluate how our system performs. We also illustrate our choices of the different tools and programming languages used in designing and developing our system and give our justifications for choosing each one. The final chapter is used to give a brief summary of the whole body of work where the major contributions of the thesis are provided. The final section of the final chapter is dedicated to the indication of future works that the researchers wished to extend but couldn't because of different constraints.

Chapter 2: Literature Review

2.1 Introduction

This chapter presents a review of concepts relevant to obtain basic understanding of the ideas of the proposed research work and concepts related to researches towards action recognition and image processing. As the proposed research is a subset of video event detection, the review presented in this section deals with basic theories important to the sphere of the proposed work; in particular, ideas on event understanding, motion and object detection, behavior understanding, human activity recognition as well as machine learning techniques, and other related issues.

2.2 Event

As mentioned earlier, shoplifting detection is a subset of event detection in which a specific event, shoplifting in our case, is attempted to be detected from a surveillance video feed. However, before we start discussing event detection and its process, it is first conventional to understand what an event is.

In the literature we surveyed, an event has been continuously defined and redefined by the corresponding authors of different works. Words like "action" [13], [14], "behavior" [15] and "activity" [16] have also been used extensively and interchangeably in the literature to describe similar concepts, while some other have used the term [17] event itself. Here, we will try to put an operational definition for the term event as used in this thesis. Throughout this paper, we use the definition given by [18] which is the following: *An event is an object built of smaller semantic units which occupy a period of time and are described using the salient aspects of video sequence inputs.*

Once again inspired by the definition given by Lavee *et al*. [18], we define a sub-event, as a part of an event that can be broken down to smaller parts that individually can be considered as events.

As an area that is interesting, challenging and one with an endless list of applications, a lot of research work is devoted to it. Event detection, also called recognition, analysis etc., is very wide and to understand it better, we can individually examine the different types witnessed so

far in the literature reviewed. The criteria of classification include, but not limited to, complexity level and environment of footage. Depending on the criteria we set, an event can be divided into different groups as follows.

2.2.1 Complexity of Event

This particular classification of the different event detection researches done is primarily based on the complexity of the activity of interest of the corresponding work. However, it cannot be for granted that recognition of less complex actions is trouble-free or effortless, as there are plenty of requirements and challenges on each distinct application. The divisions [19] are presented below:

Presence

This is the most basic and low level of the detection problem in videos. It basically checks the presence of absence of objects and humans. As the lowest level of complexity is involved here, sometimes not even grouped in the list as a type of event. Included here are general movement detection problems which are sometimes used as a trigger to start one of the latter types of events. In some industrial surveillance systems, writing to a storage starts only after presence is detected.

Body movement and gesture

Body movement is used to communicate a lot of messages in humans and other animals. It usually involves motion in hands, the head and/or other body parts. As a computer vision problem, it is usually characterized by having little complexity and happening for a very short span of time. Simple gestures as a hand waving or a human's posture can be considered in this category [20].

Action

Still considering a single person as our point of interest, this group may contain 'actions' that might include a sequence of lower-level gestures or body movement. These 'actions' usually have a longer time of happening than the previous categories. Some examples that can be mentioned here are the actions of a person walking or a person boxing.

Interaction

Interaction actions are those that are symbolized by having two or more objects of interest interact with each other. These interactions can be object-object (a robot picking a toy) or human-object (cycling or abandoned bag detection[21]). Though mostly put here in the hierarchy, works in this group might become very complex as the sequences of actions we want to recognize become higher in number and more complex in articulation.

Groups

This category contains ‘activities’ that comprise of two or more groups of objects and their interaction usually across a large space [22], [23]. Excellent examples for this are sports activities which have been gaining popularity[24], [25].

All these event understandings are usually undertaken from a video input, however, there a sizable amount of research to try to identify these ‘activities’ from single images. In this particular research, we focus on the fourth category, interaction, because our work majorly explores human-object interactions like ‘person picks object’ or ‘person puts object’.

2.2.2 Environment of event

In addition to the classification based on the complexity of the action attempted to be detected or recognized, we can also classify event detection problems based on the environments they video is set in. These categories are outdoor, indoor and mixed.

Outdoor

This category tries to detect different events in an outdoor scenery where most things are not controlled [26]. This has, per se, the most challenges than the rest because mostly no control exists on the order of things, and more seriously, there can be a lot of ‘unwanted’ background movements like cars, birds flying, tree branches waving under a breeze and the sun’s light changing illuminations of objects.

Indoor

This class contains analysis of videos from indoor scenes which are distinguished from their former counterpart by being ‘more controlled’ [27]. For example, the number of things that move are very limited and the lighting is, comparatively, way more steady than an outdoor

scene. Yes, there might be some fluctuations like when there is power outage or when someone turns out a light and the area is partially lit or the introduction of an opened window but all this are not as drastically dynamic as an outdoor scene.

Mixed or defined

The last class, is an unstructured one, where a combination of indoor and outdoor scenes exist for the analysis or when there is a defined space for the videos. A good example for the defined space is a football pitch where, at least, the area of surveillance is more delimited and organized while we can predict the common activities.

Our research focuses on the second type, indoor events, as what we want to achieve is shoplifting in different environments that are almost universally indoor.

2.3 Event Detection

Event detection, as the name implies, is the process of attempting to identify, detect or recognize an event from a specific media, usually video. The goal of event detection is to identify and localize specified spatio-temporal patterns in video, such as a person waving his or her hand [28].

As a difficult problem to tackle, many literature has been dedicated to video event detection and other video analysis topics. Different methods were employed on it from different researchers. Therefore, a researcher asks a few questions and tries to answer them accordingly in order to choose the different methods, techniques and approaches available, or to create a new one, to successfully disentangle the problem of event detection and performs a set of sub-tasks. The basic questions that call for a resolution are:

- How should we represent the object/s that needs detection and/or tracking?
- What image features to extract from the object/s?
- How to detect the object in a scene?

In the literature surveyed, each of the above questions are answered in numerous ways as the authors saw fit and usage of different methodologies is observed. We will try to discuss the various methods used on each of the issues. The answers to the above questions formalize to create a more conventional set of sub-tasks that are presented in the following sections.

In this paper, we are specifically interested in an event termed here as shoplifting. Shoplifting is the purposeful taking of a merchandise without paying from a store where it is displayed for sale or show. However, since it is really difficult to identify whether a merchandise was taken purposefully or not, we take it as an operational definition that any unpaid pocketing of a merchandise from a store to be an event of shoplifting.

2.4 Application Area of Event Detection

In the last section, we have attempted to cover the different types of event detection we have observed in the literature. It is not difficult to see the great relevance and enormous capacity the field of event detection and activity recognition provides. This piece revises the various application areas the field of event detection has been applied to.

Video Indexing and Content-based video analysis

In content-based video analysis, activity detection can be used for different applications in which understanding of the content of a video is important. These applications can be content based video summarization [25], [29], video annotation [30], [31], video similarity detection and video copy detection [32]. On a related task, as finding information in multimedia is a very difficult task and videos are the rule rather than the exception, activity detection and analysis in videos helps indexing of videos. A few applications that make use of indexing are digital museums, video browsing, search suggestions and video surveillance [33].

Human-Computer Interactions

Activity recognition, more specifically the lower levels and less complex ones like hand gesture recognition are used to interact with different devices that usually have a camera sensor. Some application are discussed here in their respective areas of application [34]. This has been observed in assistive technologies like gestures used to operate a wheelchair or where sign languages are translated to written material using these methods. In medical environments, basic gestures and facial expressions have been used as an input method for different selections, browsing through medical images of the patient in the operation theatre for instance. In the entertainment sector, applications that use video-based gesture recognition has been on the rise, especially in the gaming industry where it has been used in a variety of ways.

Surveillance

In its traditional state, a video surveillance system involves just a video camera or set of video cameras whose feed is attentively monitored by humans to make sense of what is happening in a scene. We have somehow outlived those days, however, and automatic and semi-automatic surveillance systems that are pushing the boundaries of what is possible in the area are on the rise. Applications such as suspicious event detection, abandoned bag detection and crowd counting can be a few examples [34].

2.5 Generic Architecture of an Event-Detection System

Generally speaking, any event detection system follows many or all of the following set of steps to reach its goal, loosely in this order; extracting of frames from the video, preprocessing (noise removal for example), object detection/motion detection, feature extraction and activity detection. Depending on the application of the system, object tracking and/or human tracking can be an additional task.

A general architecture of an Activity Recognition system has the components that process the input to decide to put a label to the sequence of images. These components include, but are not limited to, Preprocessing, Environment Modeling, Motion Segmentation, Object Classification, Object/Human Tracking and Behavior Understanding. Depending on the type of activity of interest (detection, recognition etc…), type of camera used (PTZ or Static, Single or Multiple), other components could be included to enhance or improve the process.

The following section discusses related literature of the subtopics and presents a brief summary of the components of an event detection process.

2.5.1 Frame Extraction

A video is nothing but a set of images, called frames, shown multiple times per second, termed the frame rate, in succession. In order to process a video for any purposes, it is usually recommended to work on individual images or frames that the video is composed of. Hence, frame extraction is the process of extracting individual frames, whole or part, from a given video input. Extracting frames is done as a straight forward process: get the frame rate of the video and export each frame as a single image. However, regardless of the ease and straight forwardness of this process, there are some issues to consider. The first and major one is

whether each frame is necessary or not considering the result needed at hand. The more the number of frames used, the better the accuracy of the result but the less the efficiency of the system as it needs to process a large number of images.

2.5.2 Preprocessing

Preprocessing is done to enhance the quality of the image in the sequence for later stages. It can also be used to make the data on hand more accessible and something simpler that can be worked on. One can also argue frame extraction to be a part of preprocessing stage when videos are the topic of subject. Video frames usually have a lot of noise due to camera quality and construction, illumination, reflections and other factors. Additionally, frames might not be of the same size if they come from two different source. Another reason why one needs preprocessing before an endeavor would be the color space of the source as opposed to the color source of the application the image is needed for. Hence, different techniques and algorithms are used to improve the quality of the image or to make it more accessible. Some processes include resizing, noise removal, de-blurring and color space conversion.

2.5.3 Motion Detection

The next stage is motion detection, also termed in some literature as Motion Segmentation, which separates foreground moving objects from the 'static' background. Most systems maintain this stage by first modeling the environment, which should be easier for a fixed camera setting where the background image is almost always the same disregarding the lighting and illumination.

In their work, Zappella *et al*. [35] have discussed different challenges of motion segmentation in video; some of which are noise, blurring, missing data from occlusion and lack of a priori knowledge. Additionally, they have discussed a set of different attributes that an object segmentation algorithm can have like handling occlusion, multiple objects, robustness and the like. Most works solve the issue of motion detection by applying three basic ‘steps’ namely, environment modeling, motion segmentation and object detection/classification.

Environment Modeling

Also known as background modeling, this step plays a crucial role as it has great impact on the efficiency of what comes after it. This is the step in the process where an ‘environment’ is

built and periodically maintained to ‘constantly’ represent a static background as opposed to moving and foreground objects.

In his thorough survey, Bouwmans [36] critically reviewed and presented a set of different background representation models. The first category includes simpler techniques and the paper termed them as basic background models; the environment modeling is done through analysis of the video over time. This analysis can be average-based [37], median-based [38] or histogram based [39]. Once a model has been constructed, each new frame is subtracted from the model to get the foreground object, of course, after adjusting by thresholding. The median background extraction method requires sorting the image series to obtain the median values and hence demands more computing time.

Another category of backgrounding modeling techniques is the statistical background modeling where statistical variables are exploited to decide which pixels belong to the foreground and which to the background image. Here, the background has been represented using Gaussian Mixture [40], Kernel density estimation [41] and the like. Such methods in time adjusts the background model for newly introduced static objects by increasing their weights against the background as they stay longer. The loophole in such algorithms is that it assumes the staticity of the Gaussian component weights between updates which, in practice, may “show multi-modal behavior”.

Yet another category of techniques for environment modeling is the use of filters to estimate the background. Pixels from the frame being examined that stray considerably from a value predicted from the background model is considered to be part of the foreground. In this category, application of filters such as the Kalman filter [42], [43] and Tchebychev filter [44], exist in the research we have examined. Though more robust to dynamic changes, these methods are not that apt for real-time applications as they have high computational complexity which weighs down heavily on the speed of the system [45].

Motion Segmentation

Motion segmentation in videos tries to detect movement of objects like humans or animals which should be a major focus of the whole event detection paradigm as only these regions of interest may be considered for recognition or detection of events.

Different techniques are used to segment motion from their "background" or “environment models” and can be grouped loosely as image/frame difference [46], [47], temporal differencing, statistical [48], [49], optical flow [50] and factorization [51][52].

Frame Difference Method

Frame difference, also termed as background subtraction is some literature, is the most common and relatively simpler technique and it computes the image difference of the current frame from the previous one or more practically, from an environment model. The output is then decided after such a computation based on a threshold to decide whether each pixel is part of 'a' foreground or 'the' background.

Various techniques include the difference of consecutive frames, the difference of the current frame with the median image [53], or with the moving average reference frame [54].

The decision for what the threshold value should be is an intricate matter as a value too low is susceptible to 'unwanted' noise while a value too high would judge a foreground object as part of the background, especially when the color and intensity of a foreground is similar to that of the corresponding (x,y) pixel of the background. Another challenge for such an approach is 'new' objects that stop for some time in a frame could be regarded as part of the background. Therefore, this method is highly affected by the quality of the background model created for it [55].

An adaptive frame differencing[56] tries to alleviate the problem of changes in background and lighting and uses a specific background model, updated every now and then, with the current frame which helps a lot for considering static objects that appear in a frame and stay for longer time frame like a new shelf or a chair introduced later in the video.

From these, we can conclude that the frame differencing method, while it could have acceptable results in fairly static indoor circumstances, it is highly ineffective where cameras are free to shoot in whichever directions and a lot of dynamic objects frequently appear. Moreover, the need to create background models every now and then weights down heavily on them.

Temporal Difference

The temporal difference techniques utilize the pixel by pixel difference of two or three consecutive frames from a video for foreground segmentation. In contrast to frame differencing, this method does not need to create background models as it uses the selected frames. This technique helps it achieve better results even on more dynamic backgrounds. However, some literature like [57] criticizes this method as having meager results in capturing each relevant pixel. It also fails to extract the shapes of non-rigid moving objects. To counter these issues and provide a more robust technique, hybrid approaches have been suggested by [58].

Statistical

On the other hand, different statistical methods are also used in different motion detection challenges. This method took its name from its technique of applying statistical manipulations on individual or group pixels to create an algorithm more robust to noise, illumination and shadow. Statistical methods have been observed in different literature attempted in different approaches like in [48] as done with Maximum A posteriori Probability (MAP), Expectation Maximization (EM) in [49] and Particle Filter (PF) in [59], [60]. Haritaoglu *et al*. [55] used a statistical model for each pixel with different intensity values taken between two frames and which are actively updated. The PF method does its job by tracking the evolution of a variable over time by constructing a sample-based representation of the probability density function[35]. The most common from the statistical methods is the Gaussian Mixture Model (GMM).

Optical Flow

Another approach to solve the same problem is the optical flow in which motion detection is based on the individual velocities of the objects in a frame. The optical flow field is basically the approximation of the movement of pixels in an image sequence [61]. Optical flow can be computed in two major ways, as dense optical flow, where the flow fields are calculated for a multitude of pixels and, sparse optical flow, which only computes flow fields for only a select set of points.

Optical flow is a widely used approach with a tremendous body of work for more than three decades and is still actively applied in the latest literature with steady frequency. The most common usage of this age-old yet very effective technique or its derivatives ranges from motion detection to feature extraction and object tracking.

This particular method has been adopted to a vast number of research areas in different flavors at different times. To mention a few, its practice has been witnessed in the medical imaging field in applications ranging from cell and organ deformations to blood flow and individual cell tracking [62][63][64], [65], in autonomous driving vehicles [66], [67], in video surveillance, in video indexing and retrieval [33][68]. Restoration of old films is another area in which optical flow methods have been employed [69], [70].

This method had a slow computation and is inefficient to detect motion in a short time as it has been attempted in different approaches at different times. However, the constant improvement of the algorithms to make them more efficient and the design involved has made it a state-of-the-art approach for many activity recognition tasks. [71]

2.5.4 Object Classification

Object classification, and activity detection are usually taken as the next stage in the process. An object can be a robot, a vehicle or a human. Human detection in a video scene of a surveillance system has wide range of applications and is an input to various systems [72].

After successful segmentation of foreground objects from the feed, the next subtask in identifying events from a video input is feature extraction followed by object classification. Basic feature extraction methods use image visual features collectively called pixel-based detection like Color Features, Texture Feature, Shape Feature and their counter parts of object classification as Color-based, Texture-based and Shape-based. It should come as no surprise that the color-based feature selection is based on the color of the objects. However, it is obvious that the apparent color of an object might not always be the same as the color being shown because of different factors like illumination and lighting.

According to Stanchev *et al*. [73], color features define subject to a particular color space or model like RGB, LUV and HSV. In their review, Zhang *et al*. [74] have tried to contrast different color descriptors. Important color features like color moments (CM), color

histograms, color coherence vector (CCV) and scalable color descriptor (SCD) are witnessed in different literature across the computer vision stream.

Color moments feature are techniques that calculate for each channel separately usually use statistical measures like standard deviation and mean. They are more often used in annotation and retrieval systems [75], [76]. Color histograms on the other hand illustrate the distribution of colors on an image. This features can also be used, especially as it is robust to some image manipulations like rotation and translation but fails to contain the spatial information of a pixel [74].

Another type of visual feature which is more common is texture feature. Two types of texture features, namely spatial texture and spectral texture, are critically reviewed on [77]. Yet another technique is the use of shape feature extraction which can also be classified into two groups as contour based and region based methods [78].

2.5.5 Human Pose Estimation

Usually in event detection system, there exists a component that tries to detect an object from images or video, the objects being anything from furniture to humans. More times than not, the object that is intended to be detected is a human and it is sometimes the case that we want to detect specific parts of a human. Human pose estimation is the process of localizing the joints of a human figure, from images or videos [79] for different purposes. Pinpointing these locations with good accuracy has a plethora of applications in the fields of computer vision. Some good examples are HCI, Assisted Living, Gesture-based applications, physical therapy and Smart Environment. As a complex problem, human pose estimation has been attempted from several direction and works like [80][81] using part detectors and pictorial structures while Yang and Ramanan [82] used mixture models and templates.

Other works have shifted their focus to 3D pose estimation as the availability of different hardware like Microsoft's Kinect camera that produces depth information along with the usual RGB image. However, the struggle to find an accurate pose estimator for humans from monocular images is still an open problem. Recent research for non-depth images have seen much better accuracy from using neural networks and have had much success in the arena [79][83].

2.5.6 Tracking

Yilmaz *et al*. [84] define tracking as *the problem of estimating the trajectory of an object in the image plane as it moves around a scene*. It is very difficult to imagine a computer vision area that does not make use of tracking and in the literature, tracking has been used in different application areas ranging from human-computer interaction to traffic monitoring and automated surveillance systems in tracking different objects principally hand and fingers, vehicles and people and humans respectively. In spite of the fact that massive resources have been allocated to research the problem of tracking, the complexity and the challenge faced by factors such as occlusion, non-articulated motion of objects and their shape deformability as well as lighting and illumination make the topic still enormously active.

Tracking of objects of interest starts from object initialization where either a human (manual) or a system (automatic) can locate the object for tracking. After the object is located, representation is usually done by modeling the moving object; and then the object is detected across successive frames where the object is represented by its model as explained in the next subsection.

Object representation

Selection of a good object representation model directly positively affects the quality and accuracy of the tracker. Selection is rightfully done according to the application and type of object we want to track. These can be shape-based, feature-based, model-based or optical flow based depending on the choices the researcher makes which are in turn based on a lot of application and object type factors. In shape-based tracking for example, point representations are more suitable for smaller objects or parts of objects intended to be tracked while objects with complex shapes are more effectively tracked by contour or silhouette representations. In their well-presented and thorough survey, X. Li *et al*. [85] state a few points that an object representation model should try to answer; what to track (box, point, silhouette etc.) shown in figure 2.2., which model is more robust, which to use for what kind of application and the like. In object tracking sense, an object can be anything of interest, a vehicle, birds flying, a baseball or a human face.

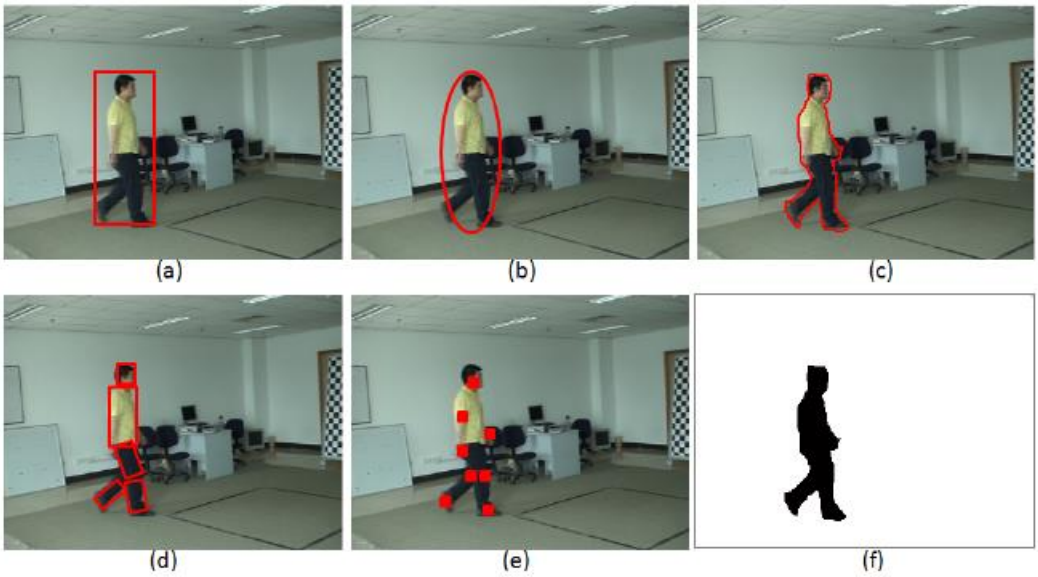

Figure 2.1: Different object tracking models [85]

(a) Bounding Box (b) Ellipse (c) Contour (d) Articulation Block

(e) Interest Point (f) Silhouette

In contrast, object representation can be based on the features rather than the shape of an object of interest. A representation feature can be categorized as global feature representation and local feature representation based on feature construction on which the survey [85] does a good job of summarizing. Global feature representations are those where the global statistical characteristics of the entity being tracked is featured and they describe an image as a whole. On the other hand, local features try to describe key points in the frame to represent the object. Some global feature examples are listed here:

Raw Pixel representation: as can be inferred from its name, this representation uses the properties of the raw pixels like their values and intensities to delineate pixels that belong to an object, as used in some works [86]. This representation is simple and fast but not robust enough when deformations capitalize.

Histogram representations: by using the histogram of an objects region, this method encodes areas belonging to an object as in researches like [87]. One major impediment for global representations is some of them have to be sharpened to handle partial occlusion. The same survey [85] continues to discuss local features that make use of an interest point to characterize the object. Some representation methods mentioned include:

Local template-based: this method undertakes the issue by decomposing the object into parts which makes it more robust to partial occlusion.

SIFT-based: SIFT, short for Scale Invariant Feature Transform, makes use of the SIFT features inside an objects area which then can be used to be matched against similar features in subsequent frames. This can be seen in the works of [88], [89] to correspond to specific regions of interest.

In conclusion, global features are easier to implement and faster while they are prone to basic changes in the shape and illumination of the object of interest. On the contrary, the local features are more robust to these changes since they better encode the local layouts. However, since they need a substantial number of local features, they are usually computationally expensive.

The third grouping is the model-based tracking, where the movement of the objects are expected to follow specific manners and prior knowledge about the object is needed [19]. While attempting to reconstruct the 3D pose for humans, Zhu *et al*. [90]used this method for tracking; while other works like [91] have used it to estimate skeleton of a human. Just as its counterpart feature-based tracking, model-based tracking also suffer from poor results when videos have substantial occlusions.

The final category to represent objects for tracking is the Optical Flow representation: a very wide technique that can also be applied for motion segmentation and full-fledged tracking, the displacement of vectors of each pixel that belongs to an object can be traced. While this method's original results have some issues like poor results in robustness and efficiency, the continued modifications applied to them [92]–[95] makes them still very competitive and a wise choice.

2.5.7 Event Modeling

Event Modeling is a very important aspect of event detection or recognition especially when the event of interest is known beforehand. Aggarwal and Ryoo [96], on their review of human activity analysis, classified event modeling as Hierarchical and Non-Hierarchical. According to this classification, non-hierarchical models are approaches that attempt to recognize events

directly from the extracted frames whereas hierarchical approaches try identifying sub-events from the frames to understand high-level events.

Another classification given by Lavee *et al*. [18] groups the task as deterministic and probabilistic event models. A more organized classification by the same authors categorizes different event models used as "pattern recognition methods", "state models" and "semantic models".

Pattern Recognition Methods

The different methods grouped under this category attend to the problem of event modeling as a traditional pattern recognition/classification problem in which an input can be classified as "Event of interest" or "Not event of interest" in its rudimentary form.

Methods under this category are known for their minimal usage of semantic knowledge in building their event classifiers. However, what they lack in the semantic knowledge, the methods try to make up for by using, sometimes fully, training data to specify the event classifiers. Nearest Neighbor, Support Vector Machine and Neural networks are examples of pattern recognition methods and are well studied and less complex in design than their counterparts mainly because they exclude semantics from the modeling.

State Models

On the contrary, more recent methods categorized under the term state models principally apply semantic knowledge of an event in space and time. These models are more flexible and adaptive than pattern recognition methods as they involve some semantic knowledge and actual event modeling and representation is encompassed unlike their former counterpart.

Semantic Event Models

This subset of models tackle the event modeling challenge as sequences of sub-events for whom relationships are described semantically by their temporal and spatial properties. While the state models discussed in the previous section use semantic knowledge as well, the semantic event models define the models in terms of the semantic rules and semantic relations between the sub-events in an event of interest.

After all the above stages, we have come up with all the necessary outputs to deduce a semantic description of the images in the video. Different approaches are used to achieve this

but the major one that catches one's attention is the sequential approach. This approaches try to defeat the challenge by analyzing sequences of features extract in earlier stages.

2.5.8 Classification Techniques

The study and computer modeling of learning processes in their multiple manifestations constitutes the topic of machine learning. In its long tenure, machine learning has served in numerous applications where it has shown great advancements. One of these is the classification problem, where a machine is shown examples of cases and it tries to learn patterns to classify new cases in their own class.

In the thorough survey, Poppe [97] presents a detailed view of different classification techniques used in action detection/recognition by categorizing them in three as *Direct Classification, Temporal State-Space Model* and *Action Detection.*

Direct Classification: try to recognize actions directly from the frames without regard to temporal information. Methods like Nearest neighbor classification (NN) have been used to recognize simple events from depth images that come from Microsoft Kinect in [98]. Another methods that falls in this category, the Support Vector Machines (SVM) are also repeatedly employed for classification of actions from video. For instance, a non-linear SVM is applied at TRECVid 2010 [99] by a group to recognize some actions like PersonRuns and Embrace.

State-Space Models: a set of states are modeled that represent an action in a temporal line. Some examples we can point here are the Hidden Markov Model (HMM) which have been very popular in the 1990s and 200s. Recent research like [100] have also utilized HMM for the event challenge of TRECVid 2011.

In recent years however, the shift towards deep neural networks, especially Convolutional Neural Networks (CNN) has been massive. As opposed to the conventional machine learning techniques where most features are designed by experts, CNNs learns the features automatically from data [101]. This makes them the optimal choice Moreover, experience has also shown that CNN are very successful when applied to 2D space and recent papers have exploited this characteristics [79].

2.6 Summary

In this chapter, we reviewed the basic flow of what constitutes an event detection from videos. Thorough examination of different existing methods to tackle each stage of the event detection procedure has been done and detailed analysis of the approaches observed has led us to a selection of methods we will use in this paper.

As naturally follows, we examined different techniques that are used to model a background environment that is sufficiently robust to basic changes like illumination, lighting and introduction of new 'static' objects in the video while at the same time having a fast enough computation to enable us to utilize it in real-time applications.

In the third section, a review is conducted on existing approaches and methods to segment movement from the background. We have attempted to understand them by grouping them in categories for which techniques sharing some commonalities are covered in the same category. Here again, the issue of accuracy vs speed of the detection plays a crucial role in choosing which methods to follow or modify that can be tailor made for the application needed.

The succeeding section is dedicated to the examination of available techniques for object tracking in different videos which plays a crucial role in different areas of application. Of the abundant research in this area, we looked at the major ones that have been extensively used and are proven to have substantial results.

The last section of the chapter discussed how events can be modeled which has an undeniably enormous effect on the final high-level output of the whole system. In that section, we inspected a set of methods that are used by different researchers and also attempted to compare each one according to their respective application areas.

Chapter 3: Related Work

3.1 Introduction

This chapter presents a number of works which are related to our research. Though there is no specific work that covers detection of shop-lifting per se, there are works that cover other event detection challenges. We will try to discuss the works we believe are closely related to the challenges in our topic.

Each section in this chapter discusses a number of related works grouped by the complexity level of the action they set out to detect or recognize. Furthermore, discussions of the goals and methods of each work is introduced, followed by a presentation of a summary and critical analysis and review of it. For the purpose of clarity, we have grouped the papers reviewed here into two categories based on the approach used to recognize the activities as Non-Hierarchical and Hierarchical.

3.2 Non-Hierarchical Approaches

As discussed in the previous chapter, non-hierarchical approaches are those that attempt to extract human activity information as a whole from the frames in a given video. Some related work belonging to this category that we have analyzed are presented as follows.

Yokoi *et al*. [102] presented at TRECVid 2009 provides an explanation of how four components are implemented to detect three events given as: E05:PersonRuns, E19:ElevatorNoEntry and E20:OpposingFlow. The components included in their system are (1) change detection (2) human detection (3) human tracking and (4) event detection. The change detection component the paper used is a combination of different techniques to make it robust against background movement and illumination changes. The human detector is created based on Co-occurrence Histograms of Oriented Gradients (CoHOGs) and Support Vector Machines (SVM). On the other hand, the human tracking that makes use of color histogram matching and Linear Estimation of Human Position is adopted by the authors. The last component, event detection, follows more straight forward approaches that analyses the results from the previous components. This paper is well presented as well as moderately mathematical and was a best run in the TRECVid 2009. There was limited or no training for

some components that might have affected the results for which the authors gave the reason of shortage of time for training.

On the other hand, Tripathi *et al*. [21] presents a method of detecting abandoned objects from surveillance video. After justifying the motivation for the research, the authors go on to describe their framework that works in three basic steps. The first step proposed is foreground object extraction which detects moving objects from a video feed. This step involves background modeling which is used as a standard for the following step of background subtraction. The last sub-task in the step in noise removal for which the authors employed erosion followed by dilation. The second step in the work is stationary object detection which used contour features from consecutive frames. The last step in the research under review is classification of objects as human or non-human stationary objects. For such a challenge, the researchers used edge based object features followed by edge based template matching. This paper is also well written with detailed explanations of each methods used. However, the testing part is not presented in a way that it can be replicated. Additionally, there are no statistical numbers that show the success rate of the experimental results even though, in our opinion, it is because of the smaller size of the sample.

Dikmen *et al*. [103] introduces and offers a detailed display of the development of three generalized systems for event detection. The systems are designed and tested for OpposingFlow, TakePicture, CellToEar, Pointing and ElevetorNoEntry tasks. The first system the authors developed is based on the principle that an event detection problem is a retrieval problem that tries to find an event of interest from a video using brute force search methods. This system uses optical flow as a motion description and Euclidean distance for measuring similarity. Frames with a specified distance from the selected image are then stored for further investigations while the rest are pruned. Afterwards, meanshift clustering is used to cluster the frames containing the 'detected events' which uses a uniform 3D rectangular kernel for efficient implementation.

As can be inferred from the work, this system basically uses a pattern recognition method to detect events and no specific modeling of event appearances exist. Though the authors stated that the results are competitive, they did not specifically put numerical figures that describe

the competitiveness. Additionally, we believe that it is inefficient to approach the problem of event detection as retrieval because of the expensive computation, especially when a media like video is involved. The authors' second system attacks the problem by narrowing the solution space by first detecting human presence and tracking. The researchers classified the events in two categories by their behavior as events requiring understanding of body movement like CellToEar and events that can be discerned from moving trajectories of a single figure, like ElevetorNoEntry. For the former task which is more complex, human detection is achieved through the use Convolutional Neural Network to detect heads and, to detect the motion of body parts for event understanding, the authors have utilized three machine learning approaches including SVM. On the other hand, for the second category of events, a rule-based method is proposed by training from data using some characteristics like location and velocity. As we have observed from the results, false positives were high while the performance was not stable. The last system proposed by the writers in which only three events are considered from the TRECVid set of tasks, applies trajectory based tracking proposed by Han *et al*. [104] to get the human motion trajectories. Training is done on manually annotated events from videos using different classifiers based on Conditional Random Fields (CRF). Then the best classifier was selected from the experimental results and put on operation in the corresponding event for which it showed superiority. The authors also assert to have come up with their own Conditional Random Field they called Latent Pose Conditional Random Field (LPCRF) which they stated can bridge the gap between the high dimensional observations and the random fields. [The new LPCRF has mixed result as compared to the other existing models]. The paper is ordinarily written and, though it rarely happens, it does have some typos. The work is at times highly mathematical and fairly presented; however, we are afraid it did not live up to our expectations. First, it promises to present a newly developed generalized system for event detection when in reality, it just attempts to solve a few events from training labeled data.

3.3 Hierarchical Approaches

As opposed to its counterpart, the hierarchical approach attempts to recognize an activity by trying to understand it as a composite of other smaller actions. This lower level actions are usually thought to be easier to model and detect than the whole event, hence, by first detecting

these shorter actions, one can build towards recognition of the more complex ones. Different researches that utilize this approach are discussed as follows:

Baxter *et al*. [105] present a new method for recognition of high level events from real-time video. The paper shows a framework with three major components. The first component detects low level objects and tracks them from the video. For the object detection, the authors used the traditional background subtraction method and foreground blobs for connected objects; on the other hand for the person tracker, a set of SIR filters are used. The paper goes further to use a hundred particles to detect temporary occlusion of the subject and to predict the whereabouts. The second component, the simple event recognition, uses the input of the first component with some basic rules to estimate different events like GroupFormed or GroupSplit. Finally, a third component to recognize complex events, has been defined. The authors modeled a complex event as a sequence of activities and put a specific sequence as a goal for detection. The complex event detector initiates only after a simple event set as a starting sequence for a complex event is detected. For the complex event detection, Baxter *et al*. used the Dynamic Bayesian Network (DBN). The paper is excellently presented and well tested. Using background subtraction method for detection is prone to errors due to small illumination changes and camera movement. Though the authors claimed an F-score of >0.92 for the person tracker, they have admitted that mis-classification of trolleys as people and other errors have been observed during the testing. The further the testing went up the framework, the less the accuracy because of the "impurities" of the inputs. Additionally, the authors have pointed out that data unavailability is the biggest hurdle in the field of surveillance and event detection.

This more recent work from Lee and Nevatia [17] attempts to detect abnormal activities from outdoor surveillance camera. The major aim of the proposed work is to detect abnormal and/or suspicious event from the video footage. Though a suspicious event can be open ended in that a list cannot get exhaustive about the events included, and one can argue to take a list of ‘normal’ events and categorize any other as ‘abnormal’, the researchers have pre-defined a set of abnormal activities. The pre-defined suspicious or abnormal events that this work attacks are *Illegal entry, Person fell down after collision* and *Line formation,* and detection of these events from the feed also depends on the location of the event. Their proposed system has

some requirements to detect the abnormal events. It depends on what is termed as a baseline surveillance system where some hardware that is capable of processing some crucial tasks in real-time and input it to their system. Some features the baseline surveillance system is expected to perform and transfer in real-time include scene calibration, environment modeling, motion segmentation and object tracking (trajectories). After this input is forwarded to the proposed system through the *track handler* module, the system implements a trajectory based low-level real-time abnormal event detection tailored to a single corresponding suspicious activity. The low-level process uses not only spatial but also temporal cues to estimate the nature of the events. If such an action is detected, the *event object handler* module requests data before-and-after the detected action suspected as abnormal. This is where the high-level semi-real-time comes into play which is responsible for processes that verify and make decisions from the low-level inputs. The modules here deal with computationally more intensive operations like human pose detection in different poses. Training based on samples from moving humans was done by extracting the blobs using background subtraction and the authors chose Joint Ranking of Granules (JRoG) as training classification algorithm. The last module in their system is the *decision maker* which collects all available information from the different cameras to inspect if the low-level detection was really a suspicious activity by a human being. The research is elegantly written and not difficult to follow. Furthermore, it does not bite off more than it can chew as practical and achievable goals are set by the authors especially for the speed of the system. The results reported by the duo for Illegal entry, Line formation and People collision 0.84, 0.8, 0.69 recall and 1.0, 1.0, 1.0 precision respectively. However, the system was not tested on publicly available data, rather on staged data prepared for evaluation purposes. Furthermore, the data it has been tested on is very small to fully persuade us of its solid results.

3.4 Summary

For surveillance event detection, the framework usually contains an object/human detector and a tracker along with an activity detector. An activity in this sense is described as a simplistic action which makes a building block of an event which can be a set or sequence of activities. There are numerous works that try to defeat the challenge of event detection from surveillance video, setting out with different goals and metrics.

In this section, we have investigated different works that strive to accomplish different event detection tasks. Initiatives like TRECVid and CVPR that attempt to present multiple tasks for researchers to bring their own version of solutions exist and make the science to take a step, if not a leap, forward. From the related work, it is evident that there is a pattern where hierarchical approaches are favored than non-hierarchical ones when the event in question can be decomposed into simpler action. We believe a shoplifting event is complex enough to be decomposed into simpler actions or sub-events that can be dealt with in a better way.

In spite of the fact that there are little works trying to detect shoplifting event, especially considering a video as the input, the literature is rich in somehow "related" issues from surveillance videos. The biggest challenge is that for each event that is proposed to be detected, a specific model has to be created and a framework has to be modified, sometimes majorly, to adapt it to the new task at hand. Moreover, the literature lacks a do-it-all framework that is generic and has acceptable accuracy for detecting a vast number of events. Each event has its own unique characteristics and traits that have to be treated considering its features and preparing models that best express the event. Hence, there is a need for a research for detection of a shoplifting event from surveillance video.

Chapter 4: Design of Real-Time Shoplifting Detection

This chapter describes our proposed system for the automatic real-time shoplifting detection. The basic architecture for the proposed system with proposed approaches and algorithms will be explained in detail. Figures and algorithms will be used to further elaborate the inner workings of the system.

4.1 Overview

As mentioned earlier, our main subject in this research is the event of shoplifting which evidently comprises some sub-events. As previously defined in this paper, shoplifting is the purposeful taking of a merchandise without paying from a store where it is displayed for sale or show. Moreover, we have stated an operational definition where any unpaid pocketing of a merchandise from a store, intentional or otherwise, to be an event of shoplifting.

We try to define what sub-events are involved in a classic shoplifting event and what are used to hint its absence. In its simplistic form, shoplifting involves the following actions, termed as sub-events here on, at its core:

- Person picks object from shelf
- Person puts object in pocket or bag

The proposed system tries to detect this sub-events and if it successfully identifies the sub-events in the order they are listed here, it will mark it as a suspicious shoplifting event and alerts the owners or surveilling officer. However, other sub-events are also detected by the system to accept or rule out the event of shoplifting. Other activities that can be considered as a sub-set of the shoplifting event are consuming of some items like food and beverages in the store without the intent of paying for them. Still others activities like wearing clothes that are displayed for purchasing without paying for them are part of shoplifting. Our research is limited to only the pocketing of items and does not intend to address the other aspects.

We did not try to address to determine if a shoplifting activity is benign or malignant as it is almost impossible to differentiate between them. The main justification for this is that the two activities are almost identical as far as visual perception is concerned and even human officers might not be able to tell apart between the two.

4.2 Shoplifting Event

It is very challenging, even for humans, to decide whether an event involves shoplifting or not. To start with, the definition of shoplifting varies across different fields and areas. However, based on the operational definition that we have remarked formerly, we tried to reconstruct what a shoplifting event contains and in what order. As far as this paper is concerned, shoplifting involves humans picking an object from a shelf, a counter or a trolley and putting it in their pocket, purse or bag. From this accord, we can try and design an architecture to handle the detection of such events from a video feed, for which we have dedicated the following section.

4.3 Shoplifting Detector Design Goals

This section explains what is expected from the proposed model whose major purpose is to detect shoplifting event from a video feed.

When we say automatic real-time shoplifting detection, we are using different adjectives implying what the system must do. In the next section, we will try to discuss the different requirements of the system we are designing.

Automatic

As the title entails, our system is expected to perform tasks automatically. But what do we mean by automatic? Throughout the literature, there are numerous works that claim automatic systems, and for obvious reasons, automaticity is very essential in topics of video and events than most other fields of study. Most literature also agree by the inherent definition of automatic and this thesis is no exception, by expecting our system to be automatic, we mean the system should operate needing little or no user interactions in the identification process.

The following features of the system should be automatic:

- Motion detection
- Human-pose estimation
- Feature Extraction
- Sub-events detection
- Alerting

Real-time

Just as the automatic tag that is appended to many event recognition systems, the real-time property is also found in a number of similar systems, and, yet again, our proposed model considers real-time processing during the event identification process. In this paper, we define real-time as soft real-time, meaning that the system's quality degrades if a deadline is not met. The detection is recommended to take place while the activity is still in progress and at worst, while the perpetrator is still in the vicinity of the shopping premises.

- The video processing unit should be able to detect movement and track objects.
- The system should alert when a shoplifting event is detected.

4.4 System Architecture

Automatic real-time shoplifting detection is the process of detecting and alerting a shoplifting event from surveillance video. Automatic shoplifting detection requires, as discussed before, different components to come up with a workable result. A general of the system is shown in the Figure 4.1: High-Level Automatic Shoplifting Detection Architecture.

The Automatic Shoplifting Event Detection system, Figure 4.1 has two major phases as can be seen from Figure 4.1, the Shoplifting Event Training and the Shoplifting Detection phase. The Shoplifting Event Training subsystem is in charge of learning the different traits of a human in a shoplifting event from a training dataset and then, to create a model which can represent these various traits. It does this by first accepting a video, applying different preprocessing techniques like extracting frames and resizing, and secondly, by applying human pose estimation techniques. It continues by extracting features from the images, these features extracted from the set of frames of a lot of video data are used to build a model that can be further used to determine if and which of the sub-events has occurred.

On the other hand, the Shoplifting Detection subsystem does all the procedures as the Shoplifting Event Training subsystem like preprocessing and feature extraction but instead of the Learning component, it has another component, the Shoplifting Event Detection component. This component uses the constructed model from the Knowledge Base along with a set of rules to detect whether a shoplifting event has occurred. If it decides that the event has

indeed occurred, it send an alert to the user. The next sections are dedicated to discussing these subsystems, components and processes in detail.

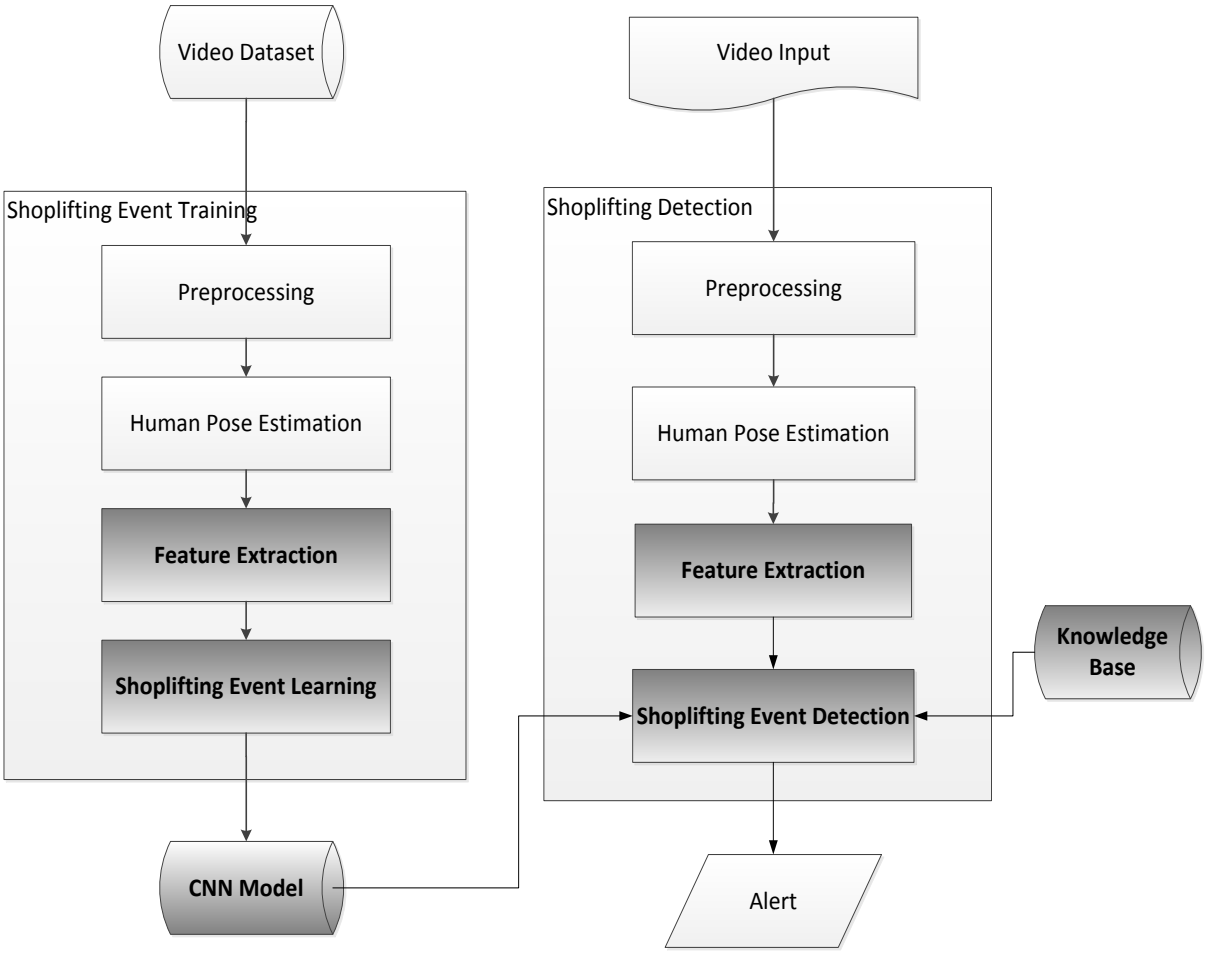

Figure 4.1: High-Level Automatic Shoplifting Detection Architecture

As can be noted from the Figure 4.1, the components that we have improved upon are shaded and are written in bold. Detailed explanations of what has been improved is further shown in their respective sections. As is usual in surveillance systems, the video feed is recorded for different purposes like reviewing and analysis. The repository is a big storage that stores the video from the surveillance cameras for a previously agreed duration of time for later retrieval and use. Since any storage is limited in capacity and the storage needs of multimedia data as video is enormous, typically only a limited amount of data is stored and the rest is flushed as time passes.

4.5 Training

As a machine learning approach is employed for the purpose of this research, the Training component is responsible for creating a model that it has learnt from the dataset provided. The training is done by applying machine learning techniques after the features are extracted from the incoming frames. Another approach, which is the trend in deep learning is to let the machine learning itself extract the features and learn from them which is the way we have selected for our work.

We have applied machine learning techniques to train a model that can be used to infer knowledge from to help us decide in the process of shoplifting detection. To tackle the problem of finding a shoplifting dataset, we have decided recording our own videos that show the event of shoplifting in action. To train this model, we have prepared our own dataset which are a set of videos that contain different subjects performing actions in them. The dataset is also use for the future to store different video snippets that it takes from the repository and inputs to the next component namely the feature extractor. This process is done periodically where the dataset requests and accepts a set of video snippets from the repository, have their features extracted and then erases them from the memory. The preparation of the starting dataset is further reported in detail in Section 5.3.

The dataset videos we have prepared were then segmented into a set of short video snippets where each one represents a class that we want the model to learn. We reviewed all the videos and took the frames where an action starts and where it ends and these were marked. We removed the actions where the action is occluded beyond recognition since it is difficult to get meaningful information out of such snippets. The next step is to preprocess these video snippets so they can be ready for the processes.

4.5.1 Preprocessing

The system generally takes a video as input, processes the images in the video and after a set of processes, output a result when appropriate. We define video as a sequence of images taken from a surveillance camera with at least 20 images, referred to as frames, per second. The video is going to be an input to the system and the frames of the video are going to be extracted so that each image can be processed. As we mentioned in the previous sections, a video usually

has around 24 frames that are taken in a single second. This particular property increases the number of frames in a video feed. For instance, a single 7 minute video contains more than 10,000 images, assuming 24fps, which makes video processing a processor-intensive task. In spite of this, basic human actions are not that fast to need that many frames per second to be observed, therefore, making some frames "superfluous". When the question of efficiency is raised, these two issues are guaranteed to be mentioned as well, namely, the fact that a video footage contains an astronomical number of frames which leads to high computational power need and the other being that we seldom need every frame for a basic human action. For these two reasons, we have decided to minimize the number of frames we process which is shown in the Frame Extraction section.

Preprocessing involves the act of tailoring an input so that it can have basic properties as can be accepted by a particular component. In our work, the tasks the Preprocessing component performs includes different activities to convert the input video into a set of images that can be processed by the Background Modeling module. These set of tasks that we perform on the input are frame extraction, color space conversion and resizing and are reflected hereafter.

Frame Extraction

As stated above, a video is composed of a sequence of frames (images) that are displayed at a fixed frame rate termed as the fps. In order to work on a video, we have to be able to get the individual images that are composed in that video. Therefore, the frame extraction is in charge of extracting the individual images from a given video and storing them on the physical storage.

The frame extractor takes the video as input, gets the frame rate of the video and extracts images from the video. Algorithm 4.1 presents how the frames of the video are extracted from a footage. After this simple step, the extracted frames are further processed to get the desired result and when the frames are ready, they are given to the next component.

```
# Frame Extraction Algorithm
Input: cap: Video

Variables: framerate, skip, counter: Int
           frame: Image
           DEFAULT_FPS = 24
           SAMPLING_RATE = 6
Output: list: List // List of Images

Begin
     frameRate = cap.getFrameRate()
     IF !frameRate  THEN
          frameRate = DEFAULT_FPS
     ENDIF
     skip = int(frameRate / SAMPLING_RATE)
     counter = 1
     WHILE !EOF
          IF int(counter % skip) == 0
               frame = cap.getVideoFrame()
          ENDIF
          counter ++;
          Add frame to list
     ENDWHILE
Return list
END
```

Algorithm 4.1: Frame Extraction

But the inevitable question is, how many frames should we utilize to infer such understanding of the basic actions a human undertakes? We have tried to show what this looks like by using a video in 30 frames per second but only sampling a specific number of frames per second. The following figures, *Figure 4.2* to *Figure 4.5*, show three consecutive frames at different frame rates.

Figure 4.2: Consecutive frames at 30 fps

Figure 4.3: Consecutive frames at 15 fps

Figure 4.4: Consecutive frames at 6 fps

Figure 4.5: Consecutive frames at 3 fps

From observations, it is easy to notice that higher frames per second have images that are almost identical to each other and hence, can be skipped in favor of maintaining efficiency. Though higher frame rates are apt in object with higher speeds like cars, or in scenarios where the actions might have fast moving objects, like ground tennis, such frame rates are superfluous in our case where shopping takes place. Though the above figures demonstrate a human in a shopping environment in a casually normal speed, it is evident to observe that even in faster acting humans, the use of a couple of frames in a single second is more than enough. Additionally, in the literature surveyed, more than half of surveillance cameras are set to capture 6 to 10 frames in a second [106]. From the above reasons, we have selected 6 frames per second as our operational fps for the research.

Color Space Conversion

Most videos in the modern world are full colored with bigger bit depth; by bit depth we mean bits per pixel. Such videos are rich in color and represent way more than their counterparts, binary and grayscale can. However, this rich presentation comes at a price of inflated memory space.

For this research, we decided that color information is not a necessity and that grayscale images are sufficient to compute what we want. Hence, we convert the extracted frame from colored (RGB) to grayscale. This minimizes the number of pixels we process by a big margin which in turn makes the system more efficient.

Resizing the Frame

Surveillance cameras have their own various resolution they record videos at. Some of them even have different resolutions that an operator can choose from with the most common being 176x120 (QCIF), 352x240 (CIF), 640x480, 704x480 (D1), 1280x720 (HD) and 1920x1080 (Full HD). The output image from videos of these range of options from different camera manufacturers and modes need to be resized to a common size in order to process effectively. For our work, we have a minimum requirement of 640x480 for which life size humans would not be too small to detect their body parts. If the surveillance camera for which the system is used has higher native video resolutions, resizing of the images is done to the resolution of 640x480.

Data Normalization

Data normalization, also known as feature scaling, is a method that we have employed to standardize the range of the dataset. This process is necessary as different raw values provide ranging outputs and we have to make sure so that the attributes contribute in a more proportionate fashion. For our specific case, this can be achieved by subtracting the mean image from each image in the set of frames we have extracted using the following formula.

$$x' = \frac{x - average(x)}{\max(x) - min}$$

4.5.2 Human Pose Estimation

Human pose estimation defeats the problem of detect the major joints of a human being from an image or a video. These joints can further be used for different image and video analysis as well as other application that are discussed in Chapter 2. For this thesis, we propose a model for identifying features that relate to a set of joints. As is obvious and can be observed from videos, shoplifting event involves the use of the hands; other body parts are the pockets that are usually around a humans' hips. The model we propose tries to extract local features from around these joints instead of computing an algorithm like optical flow estimation which is known to demand a lot of computing on the whole frame. Focusing just on a set of joints facilitates our system for two crucial reasons: the first one is that, since we are proposing a real-time application, narrowing down the pool of features to be extracted makes sure

computations do not run overboard. In other words, by reducing the parts of an image to be computed we minimize the computation needs which maximizes efficiency much more. The second reason why focusing on the joints assists us is, since the features are extracted from around the joints where the action takes place, it helps the machine learning process to be focused and become more effective.

We adopted DeepPose [79], a human pose estimation technique that makes use of deep neural networks to extract 14 joints from an image and we have modified it to give only the four joints that are applicable to our problem, the left wrist, right wrist, left hip and right hip joints. These joints are fed into the feature extractor which focuses on the areas around these joints to compute the optical flow of the sequential frames. Figure 4.6 shows the estimated human pose with the right wrist to elbow connected flow.

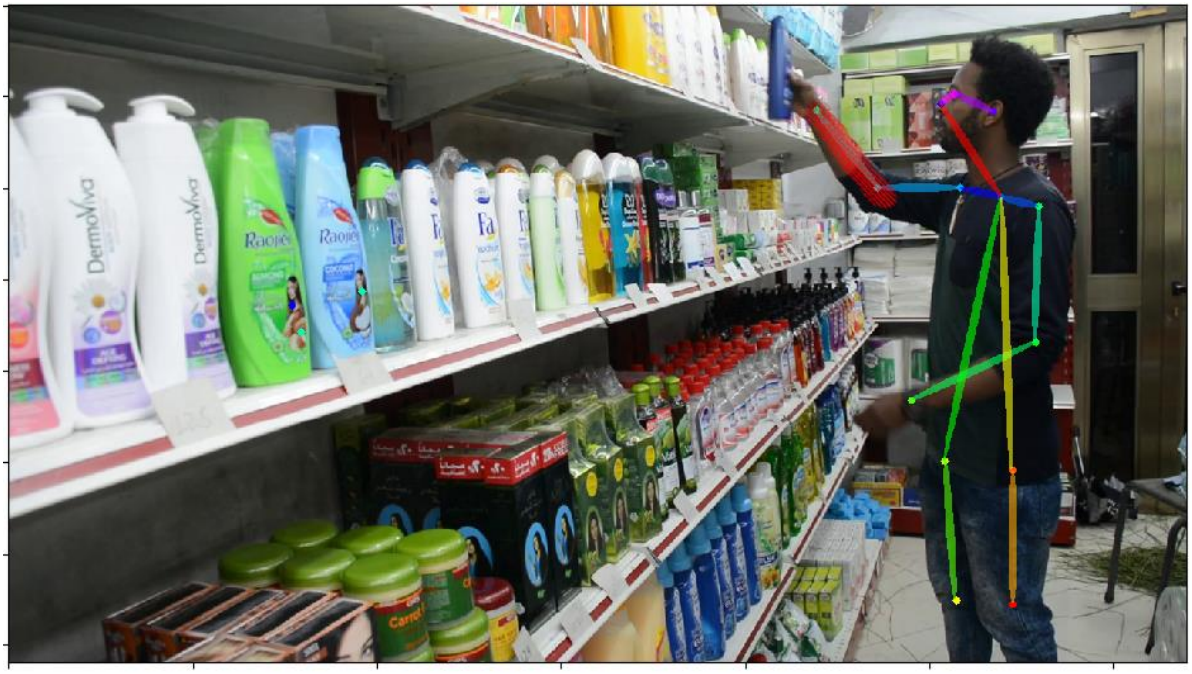

Figure 4.6: Estimated Human Pose with Flow of Right-Wrist to Right-Elbow

4.5.3 Feature Extraction

The Feature Extraction component of our model attempts to map the motion detected in the video to a set of feature vectors. We did not use any audio characteristics or signals to detect any of the actions, hence features are extracted from only the visual data of the video feeds. The choice of an optimum motion model to represent the movements in a video plays a pivotal

role on the overall performance of the system where a poor choice here reflects in all later stages acting as a bottleneck and jeopardizing the accuracy and efficiency of the whole system. From the different features a video can have, we have focused on motion as a useful feature.

Of the many options, for this thesis, optical flow estimation is used to extract features from movements in our videos which are then used to classify a movement as belonging to a specific sub-event. Optical flow method has major advantages in different action detection problems, to name a few, the fact that optical flow vectors carry temporal information and trajectories and additionally, makes training and detection more robust to different variation in view and appearance; a property that stems from the fact that optical flow is a representation invariant to appearance.

For this research, we have selected the sparse optical flow method to extract the features. This decision was made for two major reasons; the first is that dense optical flow is a compute-intensive process and it is wise to use the sparse method as the system we are designing needs near-real-time results. The second reason is that we are not that much interested in the features that are far away from our subject; the shelf that our subject is far from is irrelevant as far as shoplifting is concerned so long as we could make sure whether humans are present around it. As it accepts the location of the four joints from the previous component, the optical flow computation focuses on the areas around these joints to come up with a flow vector of features. We can then calculate the optical flow for each part of which we have an interest in (arms) to characterize the dominant direction of movement for the arms/hands, for example.

Figure 4.7: Object Put Sub-event sample

Figure 4.7 shows the sub-event of the object put sub-event as the images are ordered by time in a clockwise direction. The optical flow of these sequence of frames is computed and the extracted optical flow feature vectors are then fed into the CNN as one of the two streams and the network uses them to define a sub-event according to a set of features. A visualization of the optical flow motion history is shown in Figure 4.8. For visualization purposes, we have sampled the optical flow computation from the operational 6 frames per second to 15 frames per second. Additionally, to enhance the visualization, we have sampled 6 frames instead of 12 frames employed on the actual research. As can be observed from the Figure 4.8, the movement that is involved in the sub-event of object put is the hands.

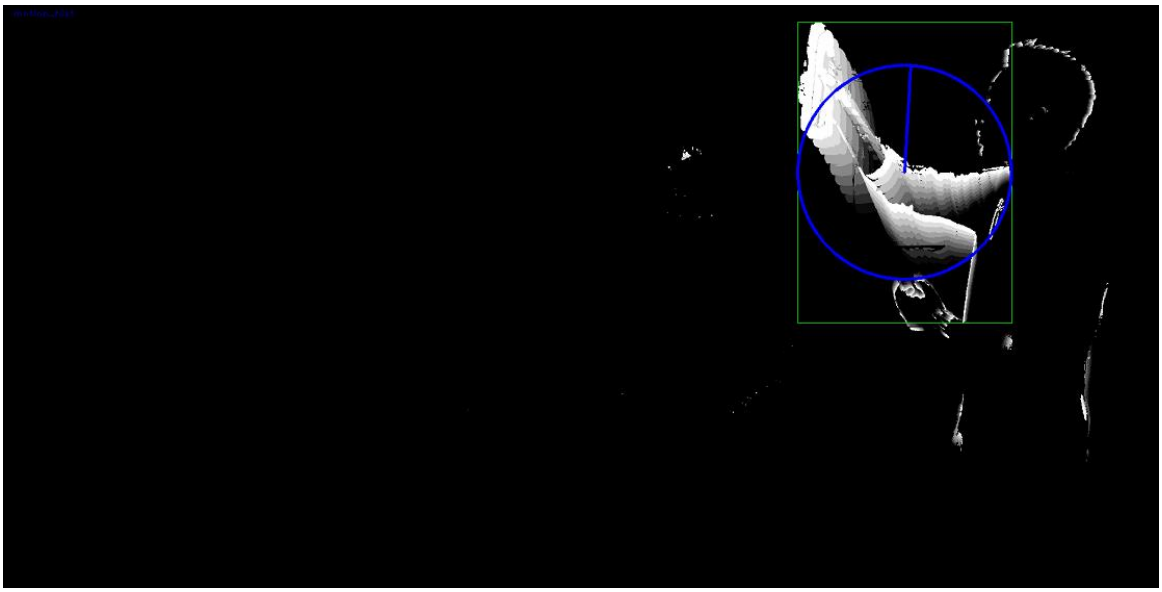

Figure 4.8: Optical Flow Motion History

Feature Selection

Following the extraction of features from the action snippets, each video clip is represented by a set of features. As one of the methods for dimension reduction, feature selection also facilitates the model to concentrate on the features which explain the phenomena of interest optimally. This is done by selecting a subset of the total features selected for the training process as some features might provide either redundant information, where one feature is enough for a group, or irrelevant information. In deep learning, as opposed to conventional machine learning techniques, feature are selected automatically by the algorithm. After each

processing layer, the feature that are more important, meaning those features which activated the next layer with higher weight, are favored against those that do not.

4.5.4 Shoplifting Event Learning

Our proposed model uses optical flow trained with CNN to learn the different movements in a specified action. Moreover, we have added a rule based algorithm to maximize the accuracy of the result. The Shoplifting Event Learning takes the inputs of the joint based flow vectors from the feature extractor and the sampled frame from the video.

After the set of features are extracted and given to this component along with the sampled frame, the event training, using a convolutional neural network takes them in a sequence of process to predicted classes. We have used an LMDB file that we generated from the datasets, an essential step in training since training especially a lot of data is a time consuming process and creating an LMDB makes this very efficient by improving the I/O performance.

Classification

After the previous two set of tasks, feature extraction and feature selection, where each time-spanned motion is represented by a flow vector, classification of the feature vectors into one of the four categories previously listed takes place. It is undeniable to be able to select the best classification algorithm is crucial to the overall effectiveness of the system. The algorithm selected in turn is heavily influenced by the type and size of corpus available as well as the output of the previous two sections in which features are extracted and selected.

For this research, we have selected the CNN machine learning approach which is a class of deep learning as it has great advantages. Unlike most machine learning techniques, CNN works directly on the raw images to extract features rather than working on selected features by an expert. This makes it a powerful instrument to learn even the features that might be unforeseen by humans which further unveils to give much better accuracy especially in complex scenarios. Additionally, the fact that CNN is able to learn with no human intervention makes it a plausible choice for later re-training after the deployment of the system with new data, so long as the CNN architecture is designed and the types of layers are chosen and how they connect and communicate to each other in the first training.

Classification is done after the features are extracted from the given data and each snippet of action video is represented by a set of feature vectors. The main goal in classification is to be able to learn to map the feature vectors to a new data provided while maintaining higher accuracy and shorter time. To ensure this, the selection of the most efficient types of features and the number of features per a sub-event has to be crafted carefully. Just as important is the design of the learning approach architecture, CNN or ConvNet in our case. Since there are numerous parameters to choose in designing a CNN architecture, making sure to select the most suited and efficient set of parameters for the network plays a key role in the overall accuracy of the model.

The CNN architecture we proposed shown in Figure 4.9 contains a model that accepts different inputs and takes them through a network of processes to classify them to a class. It is a two-stream CNN model we build upon [107], which have then been proved very successful in different action recognition tasks. As can be observed from Figure 4.9, the Sub-event CNN has two input streams, one for spatial and the other for temporal information from the videos. The spatial data is just an RGB frame that shows the person in action while the second one is an optical flow result from the 12 subsampled 'consecutive' frames, meaning 2 seconds of action. The optical flow features are those that are extracted from joint-centric methods and this makes is different from the one we build upon.

Figure 4.9 is a high-level view of the CNN and shows the different layers we have employed in our design. The first layer is the data layer which accepts the input from training or a live video from surveillance after the deployment. In our case, the CNN accepts two streams of data, followed by a convolutional layer. Our design is composed of a serious of convolutional and pooling layers on top of each other. The convolutional layer is basically composed of a filter, also known as a kernel, that slides on the whole image to convolve it acting as a feature identifiers. The pooling layer is a non-linear down-sampling layer; in our research, we have used it to add some non-linearity to the network in order to make it more adaptable to real world scenarios. Rectified Linear Units (ReLU Layers) have been utilized as a non-saturating function to also add a non-linear property. Another layer that adds to our network is the fully connected layer which helps the network detect the final output categories, in our case, the sub-events.

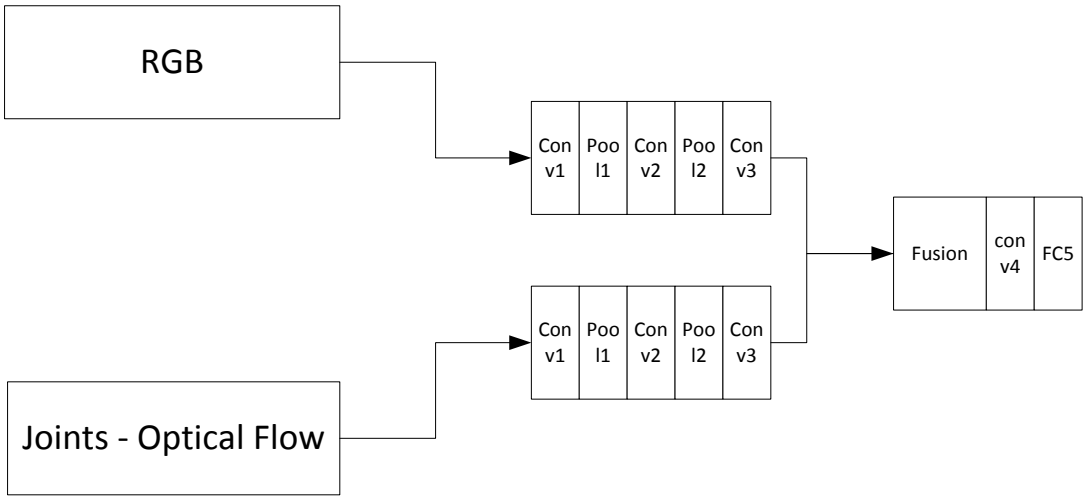

Figure 4.9: High-level architecture of the Sub-event Detection CNN

Works that have attempted to defeat the problem of action recognition use small action snippets as input to the network which have proved successful for learning atomic and simpler actions but somehow have poor results in learning longer events that might span more than the length of the fixed duration of the snippet. A very simple example can be a short football video of 10 seconds where a player scored a goal with a header. If you want to recognize this activity of "scoring with a header", it would be very difficult for a neural network to learn as a whole because it is composed of other simpler set of actions that are easier to learn. In this specific example, scoring with a header might consist of sub-events like 'jumping', 'heading the ball' and 'goal scored'. There might be cases where only the last of the three sub-event occurred but the first two did not; or only the first and the last sub-events occurred but not the 'heading the ball' one. Since it is not effective to use the whole activity as a training data, researchers have turned their faces towards instead methods where trimming the videos into smaller 'atomic' action for the network to learn is done. With this however, we are faced with yet another challenge, where to cut the videos for automatic learning? For this research, we proposed a model for trimming videos based on overlapping frames to extract motion information from. The overlapping is necessary to make sure no continuous sub-event are left in between the sampling.

4.6 Shoplifting Detection

The Shoplifting Detection subsystem performs all the processes that are described in order in the Training section, Section 4.5. The only differences between the two subsystems in discussion is that, instead of the Shoplifting Event Learning component, we have here a Shoplifting Event Detection component and an output we termed Alert. The Shoplifting Event Detection component takes a processed input and classifies it as 'shoplifting' or not,

The Shoplifting Event Detection is a major component of the Automatic Shoplifting Detection and has subcomponents that interact with each other and the outside. Included in this component are some crucial subcomponents and each of these are discussed thoroughly in the following section. Basically, the two major subcomponents we termed, in a self-explanatory nomenclature, as Sub-event Classifier and Event Classifier. Additionally, a Knowledge Base component is used along with a buffer. Let us first try to give a brief insight into each subcomponent and the Knowledge Base it communicates to.

Knowledge Base: this is a repository of knowledge that we have built from previous training data as well as model and contains patterns that can represent different actions, activities and sub-events that are of interest to our work.

Sub-event Classifier: this subcomponent is responsible for classifying frames in a scene according to the knowledge in the knowledge base as class 0, class 1, class 2 or class 3. These classifications can be further analyzed and used for event identification.

Event classifier: the event classifier subcomponent accepts input from the sub-event classifier and the buffer to classify a sequence of sub-events as shoplifting or otherwise.

Buffer: as the name suggests the buffer is used to temporarily store signals for later use by the event-classifier. It uses a simple stack data structure to perform the task.

The following architecture, Figure 4.10, shows the Event Detection component in detail:

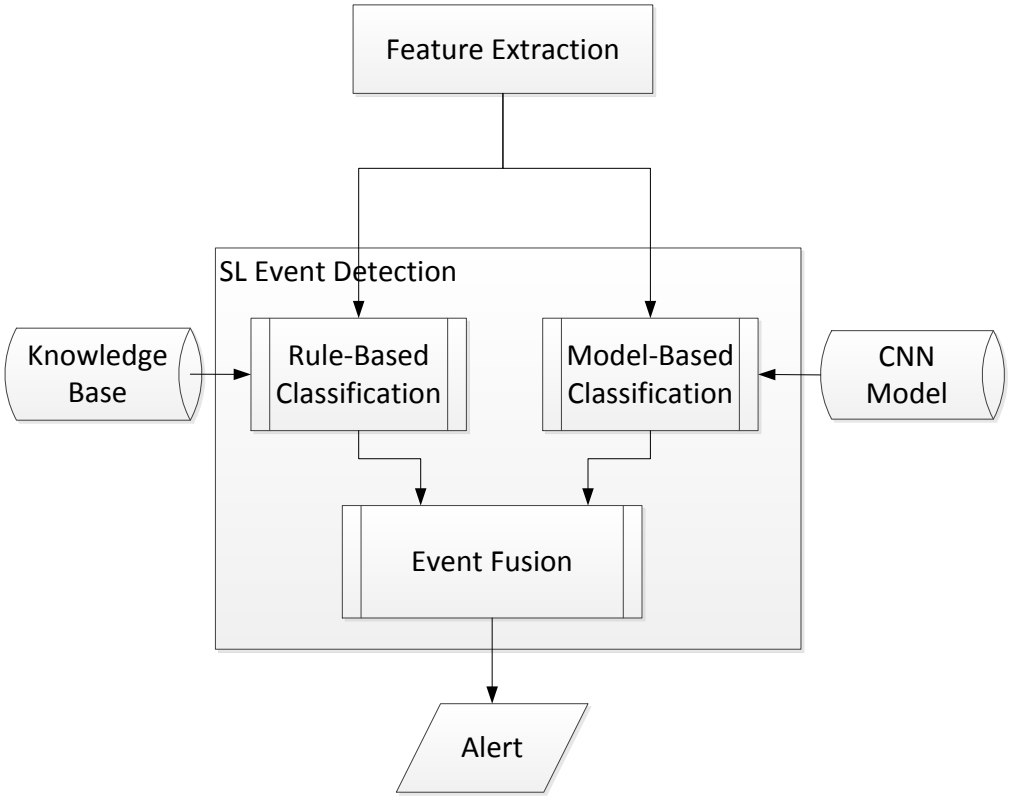

Figure 4.10: A detailed view of the Shoplifting Event Detection Component

As shown in the Figure 4.10, the Event Detection component accepts input from the previous component namely Feature Extraction and the Model-Based Classification subcomponent is initiated. The input from the previous component is the location in terms of coordinates of the subject of interest and the Shoplifting Event Detection does the process of detecting sub-events if any and deciding whether the shoplifting event has occurred or not. This is done using a set of steps that lead to an outcome and the steps are presented in Algorithm 4.2.

Each time the Shoplifting Event Detection component decides there is an event of shoplifting after processing the inputs, it sends a signal to the output, the Alert which is responsible for sending notifications in a way that could be managed by a human. This signal includes, other than the "Shoplifting detected" flag, the frame indicators for the start and end frames of the event that was marked as positive identification.

```
#### Event Fusion ####

Input: Frames: List
Output: Boolean
Begin
flags = 0
     WHILE TRUE
          flags = subEventClassifier(List)
          IF flags
               eventFuser(flags)         ##algorithm 4.4
               flags = 0
          ENDIF
     END WHILE
END
```

Algorithm 4.2: Basic algorithm for Event Fusion

The following sections are dedicated to discussing each of the subcomponents of the Shoplifting Event Detection in detail.

4.6.1 Model-Based Classification Subcomponent

Per our previous definition, an event is composed of sub-events that can be a part of a bigger event. Hence it is critical to first identify or classify a sub-event to detect an event which is already understood to be composed a specific set of activities. This is where the first subcomponent, the Model-Based Classification comes into play; we use this subcomponent to classify sub-events according to a pre-decided set of classes based on knowledge and patterns given from training.

For this purpose, we have identified a set of classes of sub-events that we shall discuss in detail:

Class 1: Object pick

This class refers to the action of a human picking any object from a shelf, a counter or a trolley. This sub-event by itself does not tell us much about the event of shoplifting because a person can pick an object from the shelf to inspect it or to put it in a trolley or for any other unforeseen

reasons. However, this action with the others, if performed in a specific set of spatio-temporal orders, can be a good indication of the event of interest.

Class 2: Object put

The second class termed as object put entails the act of putting an object by a human to a trolley, a counter or a shelf. Yet again, this action by itself cannot be considered as having any significance whatsoever to be a shoplifting event. Quite the contrary, this sub-event usually implies the opposite of shoplifting and as such is used to count out an event from being considered shoplifting and is used as such. Hence, such small details are needed as aggregate to classify an event as containing shoplifting or not.

Class 3: Hand to pocket

Hand to pocket class, as the name implies, involves the act of a human bringing hands or putting hands in his/her pocket or bag. This is the most important class and at the same time the most complex of all the classes and needs a lot of work. To begin with, this sub-event by itself can be a naïve action as a person can enter his/her pocket for a multitude of reasons like putting in or taking out different articles, a phone, keys or even an empty hand.

Class 0: Nothing happens

The sub-event classifier is responsible to classify a set of frames it takes from the previous component and classify them accordingly by comparing them with prior patterns from the knowledge base. If the classifier successfully classified any set of frames as class 1 to 3, the Event Fusion accepts the set of frames and the classified flag. Nevertheless, if the Sub-event Classifier has marked the set of frames as class 0, it does not direct any signal to the next component as this class represents ‘no action of interest has happened’.

4.6.2 Rule-Based Classification

On top of the Model-Based Classification, we have added a rule based approach to increase the accuracy of the result, which is performed by the Rule-Based Classification subcomponent. This novel approach of event modelling tries to handle the problem by first accepting the estimated human pose from previous subcomponent and it tries to estimate the proximity between the hand and the pocket of the human, which helps to corroborate or refute the decision from the Model-Based Classification component. This is done by calculating the

distance of two key human body joints that almost always get involved in the event of shoplifting which are the hands and hips. As shown in the simple Algorithm 4.3, this is done by taking a frame to inspect it and use the human-pose estimation techniques to identify the four joints in the human body, namely the left wrist, the right wrist, the left hip and the right hip joints. After these joints have been estimated, the algorithm tries to calculate the distance between these joints, and, if the distance is below a given threshold, it decides that the "hand-to-pocket" action has occurred in accordance with what was discerned from the previous subcomponent.

```
#### Hand to Pocket ####
Input: Frames: List
Output: Boolean

Begin
     WHILE frames
          currentFrame = firstFrame
          LW = detectJoint(leftWrist)
          RW = detectJoint(rightWrist)
          LH = detectJoint(leftHip)
          RH = detectJoint(rightHip)
          IF distance(LW || RW, LH || RH) < threshold
               return 1;
          ELSE
               currentFrame = nextFrame
END
```

Algorithm 4.3: Hand to pocket algorithm

4.6.3 Event Fusion

The Event Fusion subcomponent is a binary classifier that outputs "Detected" or "Not-Detected" based on the input type and order from the previous subcomponents namely, the Rule-Based Classification and the Model-Based Classification. As discussed previously, the Model-Based Classification subcomponent outputs one of the four labels, Class 0, Class 1, Class 2 or Class 3. Hence, the Event Fusion subcomponent is responsible for determining

which order of classes brings forth the “Detected” output and which does not. Additionally, from the Rule-Based Classification, it accepts an output which first could substantiate or weaken the decision. For the first task, the subcomponent uses the following Algorithm 4.4. The algorithm demonstrates how the event classifier, by using the stack data structure as a buffer, implements to output the result to the next item or ignores it if the output is “Not-Detected”. This is done by taking the individual results from the previous subcomponent and using the steps discussed to decide if the event has occurred.

```
### Event Fusion Classifier (eventFuser) ###

Input:    flags: Integer
          buffer: Stack(int)
Output: Detected: Boolean
Variables: topElement

Begin
     WHILE !flags
          Wait
     END WHILE
     topElement = stack.top()
     IF flags is class1
          IF !topElement
               push(flags)
          ELSE
               Do Nothing
     ELSEIF flags is class2
          IF topElement is class1
               pop()
          ELSE
               Do Nothing
     ELSEIF flags is class3
          IF topElement is class1
               Alert
          ELSE
               Do Nothing
     ENDIF
END
```

Algorithm 4.4: Event Fusion Classifier algorithm

Another aspect used in the Event Fusion component is the ensemble method that we employed to combine the two models to get improved results. This method uses weighted voting mechanism to come up to a conclusion. A weighted voting system, unlike the standard majority voting, favors the 'better' model with some predetermined weight over the other.

As soon as a decision is made by the system of the occurrence of a shoplifting event, a signal is sent from the Event Detection component as an output which includes the start and end frame numbers. These numbers are used for two different purposes: first, all the frames starting from the start frame number up to the end frame number are displayed in a small window. Secondly, the frames numbers are sent to the repository to be marked for later retrieval for further training.

From common knowledge, alerting by sounding alarms for the whole store as is done in the case of RFID tags, we felt, is not an intelligent decision as it might be possible that we have false positives. This type of alerting is not wise for two reasons: first, it puts direct confrontations of the staff with the culprit for which they might not be trained and second, in case of false positives, creates a psychological discomfort on the honest customers. Instead, our alerting uses dialog boxes and sounds that can only be heard by operators to bring the situation to the attention of the operators.

The operators are also shown the set of frames that made the alert trigger by a sub-windows of the set of sub-events whose aggregate has been flagged as a suspected shoplifting. After reviewing the video snippet, the operator confirms or reject the action as a true shoplifting event. In both cases, the frames are taken to the repository where three things are written: starting frame, end frame and flag; where the flag indicates either true positive or false positive. The operators can then take whatever actions necessary to notify the responsible bodies about the event.

4.7 Summary

In this chapter, we discuss the core concepts and components underlying the system we have developed, each component is presented in detail to show its inner workings as well as its interaction with its counterparts. Moreover, the chapter attempts to show where the major contributions of the thesis work lies and how it fits in the present body of research.

We have set the goals of our design and the expectations we have from our system. Shoplifting, as an event, is defined, described and explained where we also gave operation definitions to it. Section 4.4 discusses the Shoplifting Detector system starting from the architecture down to each component. The type of input, data usage policies as well as the starting dataset used to train a model is reviewed and discussed. In the same section, we continued to discuss how preprocessing is done, selection of the different methods and approaches used as well as novel algorithms to make the reader understand the inner workings of our system. We have also tried to show our contributions in the general architecture by explaining our original ideas by the support of figures and easy to understand high-level algorithms. Specifically, we have shown how we have modeled the event of shoplifting using a hierarchical approach that included simpler sub-events. Later, optical flow along with CNN were used to design the Shoplifting Event Detection System. The inner working of the CNN network, the general work flow of the system as well as the types of output the system gives are demonstrated.

Chapter 5: Experiment

5.1 Introduction

This chapter presents the results and discussions on the work carried out in this thesis. This work, as mentioned earlier, has focused on solving the issue of shoplifting detection from surveillance cameras. In this chapter, the statistical and empirical aspects of the results are discussed with no dedicated space to theoretical discourse. The outcomes of the proposed solution of the detector is discussed as taken from the experiments we have conducted. Furthermore, the evaluation of the designed system is also thoroughly and closely reviewed. The ensuing results of the experimentation carried out, along with the environments in which the system was tested, are presented in the subsequent sections.

5.2 Development Tools and Experimental Environment

5.2.1 Tools and Programming Languages

This subsection states the different tools used in the development process as well as the computing environments in which the testing was undertaken.

In developing what has been proposed as a solution to the problem in discussion, we have used a number of tools. The recurrent programming language we have used is Python with which we have implemented most of our algorithms.

We used Python as the major development programming language because of the various benefits of the languages. First of all, code written in python has an outstanding readability which in turn gives rise to its undemanding maintainability. Secondly, Python provides an extensive and robust set of standard libraries in a number of areas that can be used with ease along with other open-source tools. Additionally, the language is compatible with the major platforms and systems that are on the market which makes code written in Python very portable.

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library which was designed specifically with image processing in mind, hence its selection. The library has an abundance of computer vision functions and larger community than its candidate alternative, Matlab. Moreover, functions used from this

library are much faster than their counterparts and many of these functions are written for and implemented on GPUs. These qualities of OpenCV make it the right choice for an endeavor like ours which needs speed and efficiency as real-time requirements are expected.

The Caffe deep learning framework developed by the Berkeley Vision and Learning Center at University of California, Berkeley was employed as a backbone of the solution to human pose estimation problem and our CNN. We have selected the Caffe framework for its reliable modularity, speed and decent support in the community.

5.2.2 Experimental Setup

All the experiments were done on a laptop with the following capabilities:

Table 5.1: Environment the Machine Experiments were Performed on

OS	Ubuntu 16.04 LTS
Architecture	64 Bit
Processor	Intel Core i7-7700 HQ
Processor Speed	2.8 GHz
Cores	Quad Core, 8 Logical cores
Graphics Card	GeForce GTX 1050 Ti

5.2.3 Prototype Development

We have developed a prototype to verify the proposed solution works according to our expectations. The prototype we have developed accepts a stream of video data, extracts the frames with a stride that enables it to sample 6 frames per second. As soon as these frames are extracted, the other preprocessing tasks are performed like resizing and color space conversion as well as data normalization. Once the frames are preprocessed, the feature extraction module is activated which extracts the motion features of the frames and save it in a .flo file format. This is what we feed to the network in one of the streams, the other being the raw frames.

As we have mentioned, python is used for most of the programming development along with the OpenCV library. Here is a code snippet from the frame extraction module of the system.

```
import cv2
import math

videoFile = "vid1.mp4"
imagesFolder     = "./frames/" # Directory to put images
skip = 5
cap = cv2.VideoCapture(videoFile)
frameRate = cap.get(skip) #Get frame rate
count = 0
num = 0
video_length = int(cap.get(cv2.CAP_PROP_FRAME_COUNT)) - 1
while True:
    filename = imagesFolder + 'image_' + str(num) + '.jpg'
    ret, frame = cap.read()
    if count % skip == 0:
     cv2.imwrite(filename, frame)
     num = num + 1
    count = count + 1
    if count > video_length:
        Break
        cap.release()
```

Source Code 5.1: Frame Extraction Source Code

Some decisions we have made during implementation are the threshold value for the hand to pocket algorithm which, from observation, we have found to be 0 pixels, meaning that there has to be a physical connection between the joints for algorithm to return TRUE.

Another implementation decision we have taken was in the weighted voting mechanism, and for this, we have selected the weights for the Class 1 – ObjectPick to be 2, Class 3 – HandtoPocket to be 1 and Rule-based output to be 1. Hence, if the some of these weights is 3 or more, the alert is triggered.

In the design of the CNN, we have made different choices, parameters and hyper-parameters to get the final model as an output. Here is the data layer from the CNN that we have designed to learn from the video.

```
#Data Layer of the CNN
layer {
  name: "data"
  type: "VideoData"
  top: "data"
  top: "label"
  video_data_param {
    source: "examples/train_shop/shoplift_train_split1.txt"
    batch_size: 8
    new_height: 480
    new_width: 640
    new_length: 12
    shuffle: true
  }
```

Source Code 5.2: Data Layer of the CNN

The data layer is responsible for accepting the data for training or testing, the batch size defines how many number of instances the network should take, and the length determines the number of frames that are processed. The shuffle parameter, when set to true, forces the network to shuffle the inputs to avoid a common problem of early weight imbalance to the data that is presented at the start of the dataset as many dataset might be setup in some order.

5.3 Dataset Preparation

Dataset preparation is the process of collecting and organizing data for use primarily for analysis, training and testing purposes. Though event detection system that makes use of machine learning requires a large training dataset, preparing such large quantities of data is very expensive and usually needs involvement of a lot of experts. It is a fact that such big datasets have been prepared and made available public for the research community and they are quite essential when a researcher can make use of them; however, such datasets usually

include only a subset of events that are frequently dealt with in the action detection category like ‘Person Running” or “Human Playing Guitar”. For our specific case, we have not found any publicly available dataset that can represent the actions and events representing the ones we are interested in, and, it is for this reason, that we have decided to prepare our own dataset by staging a couple of people in a realistic shopping scenario.

For this thesis, we have prepared our own dataset that we used to train our models and also to test the validity and accuracy of the system. The following set of paragraphs report the process we undertook to prepare the dataset.

The camera we have made use of is a Nikon D7100 DSLR which is capable of shooting videos at Full HD (1920x1080) at up to 60 frames per second with other multitude of options. For the purpose of this thesis, we have recorded videos at a resolution of 640x480p at 24fps. We have chosen this resolution

The dataset we prepared consists of 42 videos, each of which has a duration of 1 to 3 minutes of sequential footage. Contained in each video, are a number of subjects performing a set of actions which makes hundreds of total number of actions in the whole dataset. The total number of subjects that participated in the videos are 14, each of which appear in three videos. Each video clip has the actions of “object pick”, “object put” and “hand to pocket” a number of times. The subjects were told to move freely and to make the actions as real as best they could. Some screenshots of the dataset is shown in Figure 5.1 to Figure 5.3.

Figure 5.1: Dataset Screenshot 1

Figure 5.2: Dataset Screenshot 2

Figure 5.3: Dataset Screenshot 3

After the videos have been shot, we trimmed the videos into smaller snippets ranging from 3 seconds to 10 seconds, each containing one sub-event. The total number of video snippets we have trimmed and collected reached 311. Of these videos, we have excluded some which did not qualify for our criteria as some of them had major occlusions, the ones we have used are summarized in the table below.

Table 5.2: Dataset Clips for each Sub-event Class

Video Label	Number of Clips (Training)	Number of clips (Testing)	Total Number of Clips
Object Pick	47	21	68
Object Put	52	22	74
Hand-to-Pocket	35	16	51
Do Nothing	65	27	92
Total	199	86	285

We have numbered the subjects 1 through 14 and we have used the first 9 subjects which make up for two third of the total 42 videos for training purposes and the last 5 subjects or 15 videos for testing purposes.

5.4 Evaluation

Once we have developed a prototype, we evaluated the accuracy of the implemented algorithms and tried to assess the whole system these algorithms and components integrate to create. The experiments are conducted on the video snippets we have recorded for the purpose of this particular research. We centered on evaluating the effectiveness of two major tasks of the system, effectiveness of detecting a sub-event and effectiveness of detecting the event as a whole. In the latter, we tried to assess the difference when the Rule-Based classification is added and when it is not. The evaluation method employed is discussed in the next section.

5.4.1 Evaluation of Sub-event Classification

As presented in Chapter 4, we have created four classes of sub-event namely, Object Pick (Class 1), Object Put (Class 2), Hand-to-Pocket (Class 3) and Do Nothing (Class 0). The following Confusion Matrix from Table 5.3 shows the ground truth and prediction of each video clips tested.

Table 5.3: Confusion Matrix for the Four Sub-event Classes

Predicted / **Ground Truth**	**Class 1**	**Class 2**	**Class 3**	**Class 0**
Class 1	10	4	5	2
Class 2	4	11	3	4
Class 3	4	1	9	2
Class 0	4	2	4	17

Once the Confusion Matrix is done, it has all the things necessary for us to apply interpretations of performance measures. For the purpose of evaluating the effectiveness of

our model, we used the precision and recall of each of the sub-event classes we have attempted detect with our models. We have selected precision and recall as an evaluation metrics as they are simple to understand but serve an important purpose of giving us a critical analysis of the model tested.

Recall

Recall is the ratio of all correctly classified positive observations to the total predicted positive observations. We have computed the recall for each sub-event class as shown here:

$$Recall\ for\ Class1 = \frac{TP\ of\ Class1}{Total\ Ground\ Truth\ of\ Class1} = \frac{10}{21} = 47.6\%$$

$$Recall\ for\ Class2 = \frac{TP\ of\ Class2}{Total\ Ground\ Truth\ of\ Class2} = \frac{11}{22} = 50\%$$

$$Recall\ for\ Class3 = \frac{TP\ of\ Class3}{Total\ Ground\ Truth\ of\ Class3} = \frac{9}{16} = 56.2\%$$

$$Recall\ for\ Class0 = \frac{TP\ of\ Class0}{Total\ Ground\ Truth\ of\ Class0} = \frac{17}{27} = 63\%$$

Precision

We have also computed the precision of each sub-event classes in our classification task. Precision is a measure of performance defined as the ratio of all correctly classified positive observations and the total ground truth positive observations. The precision for each of the sub-event classes we have tested are shown here:

$$Precision\ for\ Class1 = \frac{TP\ of\ Class1}{Total\ Predicted\ Class1} = \frac{10}{22} = 45.5\%$$

$$Precision\ for\ Class2 = \frac{TP\ of\ Class2}{Total\ Predicted\ Class2} = \frac{11}{18} = 61\%$$

$$Precision\ for\ Class3 = \frac{TP\ of\ Class3}{Total\ Predicted\ Class3} = \frac{9}{21} = 42.8\%$$

$$Precision\ for\ Class0 = \frac{TP\ of\ Class0}{Total\ Predicted\ Class0} = \frac{17}{25} = 68\%$$

5.4.2 Evaluation of the Event Detection

As has been presented in Chapter 4, our event modeling consists of a semantic model where a shoplifting event is described by the different sub-events and their temporal relationships. Hence, in this section, we have evaluated the performance of the whole shoplifting event detection system. Similar to the above section, we have employed similar measures of effectiveness to this task where Confusion Matrix is shown along with computations for the Precision and Recall of the event.

Table 5.4: Confusion Matrix for Shoplifting Event

Predicted / **Ground Truth**	**Shoplift**	**No Shoplift**
Shoplift	6	5
No Shoplift	4	4

Recall

As a general frame work, we evaluated the recall for the event detection as a whole from the testing we have performed on each individual clips:

$$Recall\ for\ ED = \frac{TP\ of\ ED}{Total\ Predicted\ ED} = \frac{6}{11} = 54.5\%$$

Precision

As discussed above, the precision can also be calculated from the data in the Confusion Matrix as:

$$Recall\ for\ ED = \frac{TP\ of\ ED}{Total\ Predicted\ ED} = \frac{6}{10} = 60\%$$

As can be observed from the above results, the system performs well considering the fact that the training dataset was too small. From the results, it can be inferred that the automatic real-time shoplifting system is very usable, and with more training, accuracy can be increased. Additionally, users can manage to tradeoff between false positive and false positive values by weighing of the Rule-Base Classification as well as a specific sub-event.

5.5 Discussions

The evaluation of the prototype we developed to show the viability of the proposed work has satisfactory results as we have tried to show in the previous sub-section. This result, when compared to other works, is summarized in the next Table 5.5.

Table 5.5: Comparison to Other Works

	Environment	Complexity	Recognition approach	Actions	Results
Tran *et al.*[108]	Outdoor	Action	Non-Hierarchical	Walking; Running	Recall: 0.78
Baxter *et al.* [105]	Outdoor	Groups	Hierarchical	GroupForm; GroupSplit	F-Score > 0.92
Lee and Nevatia [17]	Indoor	Interaction	Hierarchical	AbnormalEvent	Recall: 0.84, 0.8, 0.64
The Proposed Work	Indoor	Interaction	Hierarchical	Shoplifting	Recall: 0.5, 0.56, 0.63

Chapter 6: Conclusion and Future Work

6.1 Introduction

This final chapter of the documents has three sections. In the first section, we provide a brief summary of the activities undertaken in the course of this research work. A review of the research questions and the answers that the output of the thesis has provided is presented. In the second section of the chapter, the contributions and achievements of the research work are outlined to show the importance of the work we have so far achieved. In the final section of the chapter, future works that we were not able to extend because of different factors are specified that can lead to the improvement of the work's different aspects.

6.2 Conclusion

In this thesis, real-time event detection model is developed and discussed that takes surveillance video data and predicts the event of shoplifting. In Chapter 2, we try to delve into the science that deals with basic theories important to the sphere of the proposed work; in particular, ideas on event understanding, motion and object detection, behavior understanding and human activity recognition. This helps us in identifying the methods, techniques and approaches other researchers have use to solve similar problems. In Chapter 3, we deeply analyze works of similar nature to our thesis and review them accordingly. Moreover, the research gap that we are attempting to fill has been identified which justifies the need for this thesis.

The problem of shoplifting can be reduced by the introduction of technological advances in the security of retail stores, super markets and other stores. Human operators are usually assigned to this task but there are a lot of shortcomings; this thesis shows we can use machines to detect this act from surveillance video. This is achieved because the novel CNN based model we have created can learn the event we modeled and can warn personnel that monitor the cameras. The hybrid model of shoplifting detection can be an important tool to owners that already have surveillance cameras installed in their stores.

The proposed model has two major parts, a Model-Based Classification that has learned the basic features of the sub-events from human joint areas in the event we modeled. These features are used to detect when a new observation containing this deed occurs in the view of the surveillance camera. Additional to the Model-Based Classification, the Rule-Based Classification we proposed substantiates the findings from the former to improve the accuracy of the classification.

6.3 Contributions of this Work

In general, this research work has contributes a two phase model for identifying a shoplifting event. The specific contribution of this work are:

- A modeling of shoplifting event
- A joint-based feature extraction approach for detecting shoplifting
- A CNN based network architecture that is optimal to learn the different sub-events modelled in the work that add up to a shoplifting event
- A rule-based model to support the decision about the occurrence of a shoplifting event
- Hybrid method of a rule-based and training model-based system architecture to detect a shoplifting action

6.4 Future Work

This research work explores an area in the sea of event detection which can be further extended for coverage, accuracy and functionality. As this work has been designed and implemented with single subjects in the purview of the surveillance camera, it has some limitations. Future research can be directed towards attempting extending the work to take into account the existence of multiple persons in the same frame as might be the case in real life scenarios. Another significant future work area could be broadening the work by the addition of PTZ surveillance cameras that are usually ubiquitous in the surveillance industry.

Incorporating components to handle feeds from multiple cameras that view the same thing from different angles can be an appealing future work as long as latency is taken into account.

Furthermore, this work can be extended to handle occlusion to better increase its accuracy. To summarize the future work, improvements can be added to:

- Handle occlusion to detect joints and actions even when part of the body in occluded by another entity
- Add multi-person capability to detect efficiently when more than one person exists in a single frame
- Incorporate multiple cameras, accepting and integration of a scene from different cameras and angles while effectively fusing the information
- Handle non-static cameras, support for non-static cameras where improvements can be made to map motion features even when camera movement occurs.

References

[1] R. T. Collins, A. J. Lipton, and T. Kanade, *Introduction to the special section on video surveillance*, vol. 22, no. 8. 2000.

[2] The National Association for Shoplifting Prevention, "*Shoplifting Statistics.*" [Online]. Available: http://www.shopliftingprevention.org/what-we-do/learning-resource-center/statistics/.

[3] F. Jiang, J. Yuan, S. A. Tsaftaris, and A. K. Katsaggelos, "*Anomalous video event detection using spatiotemporal context,*" *Comput. Vis. Image Underst.*, vol. 115, no. 3, pp. 323–333, 2011.

[4] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh, "*Activity Recognition and Abnormality Detection with,*" *Proc. 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, IEEE, Washington, D. C.*, vol. 1, pp. 838–845, 2005.

[5] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "*Actions as space-time shapes,*" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, 2007.

[6] Y. Ke, R. Sukthankar, and M. Hebert, "*Spatio-temporal shape and ow correlation for action recognition,*" *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1–8, 2007.

[7] M. Tien, Y. Wang, and C. Chou, "*EVENT DETECTION IN TENNIS MATCHES BASED ON VIDEO DATA MINING,*" pp. 1477–1480, 2008.

[8] W. L. Lu and J. J. Little, "*Simultaneous tracking and action recognition using the PCA-HOG descriptor,*" *Third Can. Conf. Comput. Robot Vision, CRV 2006*, vol. 2006, 2006.

[9] Y. Luo, T. Wu, and J. Hwang, "*Object-based analysis and interpretation of human motion in sports video sequences by dynamic bayesian networks,*" vol. 92, pp. 196–216, 2003.

[10] R. Bodor, B. Jackson, and N. Papanikolopoulos, "*Vision-Based Human Tracking and Activity Recognition,*" *11th Mediterr. Conf. Control Autom. - MED'03.*, pp. 18–20, 2003.

[11] R. R. and M. H. C. Lakshmi Devasena, "*Video Surveillance Systems - A Survey*," *IJCSI - Int. J. Comput. Sci.*, vol. 8, no. 8, p. 8, 2011.

[12] P. Over, G. Awad, J. Fiscus, B. Antonishek, and G. Qu, "*TRECVID 2014 -- An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics*," *TRECVID 2014 Summ.*, pp. 1–34, 2014.

[13] S. Ali, A. Basharat, and M. Shah, "*Chaotic invariants for human action recognition*," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 2007, pp. 1–8.

[14] J. Yuan, Z. Liu, and Y. Wu, "*Discriminative video pattern search for efficient action detection*," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1728–1743, 2011.

[15] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "*Behavior recognition via sparse spatio-temporal features*," in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, 2005, pp. 65–72.

[16] N. T. Nguyen, D. Q. Phung, S. Venkatesh, and H. Bui, "*Learning and detecting activities from movement trajectories using the hierarchical hidden Markov model*," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, vol. 2, pp. 955–960.

[17] S. C. Lee and R. Nevatia, "*Hierarchical abnormal event detection by real time and semi-real time multi-tasking video surveillance system*," *Mach. Vis. Appl.*, vol. 25, no. 1, pp. 133–143, 2014.

[18] G. Lavee, E. Rivlin, and M. Rudzsky, "*Understanding video events: A survey of methods for automatic interpretation of semantic occurrences in video*," *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.*, vol. 39, no. 5, pp. 489–504, 2009.

[19] S. Vishwakarma and A. Agrawal, "*A survey on activity recognition and behavior understanding in video surveillance*," pp. 983–1009, 2013.

[20] O. Patsadu, C. Nukoolkit, and B. Watanapa, "*Human Gesture Recognition Using Kinect Camera*," pp. 28–32, 2012.

[21] R. K. Tripathi, A. S. Jalal, and C. Bhatnagar, "*A framework for abandoned object detection from video surveillance*," in *Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2013 Fourth National Conference on*, 2013, pp. 1–4.

[22] W. Lin, M. T. Sun, R. Poovendran, and Z. Zhang, "*Group event detection with a varying number of group members for video surveillance*," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 8, pp. 1057–1067, 2010.

[23] W. Lin, M.-T. Sun, R. Poovendran, and Z. Zhang, "*Group event detection for video surveillance*," *2009 IEEE Int. Symp. Circuits Syst.*, pp. 2830–2833, 2009.

[24] A. D. A. Ara, "*An Overview of Automatic Event Detection in Soccer Matches*," pp. 31–38, 2010.

[25] Y. S. Khan, "*Video Summarization : Survey on Event Detection and Summarization in Soccer Videos*," vol. 6, no. 11, pp. 256–259, 2015.

[26] B. Lei and L. Q. Xu, "*Real-time outdoor video surveillance with robust foreground extraction and object tracking via multi-state transition management*," *Pattern Recognit. Lett.*, vol. 27, no. 15, pp. 1816–1825, 2006.

[27] D. Tsai and S. Lai, "*Independent Component Analysis-Based Background Subtraction for Indoor Surveillance*," no. March, 2014.

[28] Y. Ke, R. Sukthankar, and M. Hebert, "*Event detection in crowded videos*," *Proc. IEEE Int. Conf. Comput. Vis.*, 2007.

[29] A. G. Money and H. Agius, "*Video summarisation: A conceptual framework and survey of the state of the art*," *J. Vis. Commun. Image Represent.*, vol. 19, no. 2, pp. 121–143, 2008.

[30] S. Bianco, G. Ciocca, P. Napoletano, and R. Schettini, "*An interactive tool for manual,*

semi-automatic and automatic video annotation," *Comput. Vis. Image Underst.*, vol. 131, pp. 88–99, 2015.

[31] R. Poppe, "*Vision-based human motion analysis: An overview*," *Comput. Vis. Image Underst.*, vol. 108, no. 1–2, pp. 4–18, 2007.

[32] J. M. Thomas, "*A Robust And Fast Video Copy Detection System Using Spatio-Temporal Features*," vol. 2, no. 1, pp. 27–33, 2012.

[33] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "*A survey on visual content-based video indexing and retrieval*," *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.*, vol. 41, no. 6, pp. 797–819, 2011.

[34] S. R. Ke, H. Thuc, Y. J. Lee, J. N. Hwang, J. H. Yoo, and K. H. Choi, *A Review on Video-Based Human Activity Recognition*, vol. 2. 2013.

[35] L. Zappella, X. Lladó, and J. Salvi, "*Motion segmentation: A review*," *Front. Artif. Intell. Appl.*, vol. 184, no. 1, pp. 398–407, 2008.

[36] T. Bouwmans, "*Background subtraction for visual surveillance: A fuzzy approach*," *Handb. soft Comput. video Surveill.*, vol. 5, pp. 103–138, 2012.

[37] B. Lee and M. Hedley, "*Background estimation for video surveillance*," 2002.

[38] J. Zheng, Y. Wang, N. L. Nihan, and M. E. Hallenbeck, "*Detecting cycle failures at signalized intersections using video image processing*," *Comput. Civ. Infrastruct. Eng.*, vol. 21, no. 6, pp. 425–435, 2006.

[39] J. Zheng, Y. Wang, N. Nihan, and M. Hallenbeck, "*Extracting roadway background image: Mode-based approach*," *Transp. Res. Rec. J. Transp. Res. Board*, no. 1944, pp. 82–88, 2006.

[40] Z. Zivkovic, "*Improved adaptive Gaussian mixture model for background subtraction*," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 2004, vol. 2, pp. 28–31.

[41] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "*Background and*

foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proc. IEEE*, vol. 90, no. 7, pp. 1151–1163, 2002.

[42] S. Messelodi, C. M. Modena, N. Segata, and M. Zanin, "*A Kalman filter based background updating algorithm robust to sharp illumination changes*," in *International Conference on Image Analysis and Processing*, 2005, pp. 163–170.

[43] J. Zhong and others, "*Segmenting foreground objects from a dynamic textured background via a robust kalman filter*," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 2003, pp. 44–50.

[44] R. Chang, T. Gandhi, and M. M. Trivedi, "*Vision modules for a multi-sensory bridge monitoring approach*," in *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, 2004, pp. 971–976.

[45] M. Shah, J. D. Deng, and B. J. Woodford, "*Video background modeling: recent approaches, issues and our proposed techniques*," *Mach. Vis. Appl.*, vol. 25, no. 5, pp. 1105–1119, 2014.

[46] A. Cavallaro, O. Steiger, and T. Ebrahimi, "*Tracking video objects in cluttered background*," vol. 15, no. 4, pp. 1–10, 2005.

[47] F. Cheng and Y. Chen, "*Real time multiple objects tracking and identification based on discrete wavelet transform*," vol. 39, pp. 1126–1139, 2006.

[48] H. Shen, L. Zhang, B. Huang, and P. Li, "*A MAP Approach for Joint Motion Estimation , Segmentation , and Super Resolution*," vol. 16, no. 2, pp. 479–490, 2007.

[49] R. Stolkin, A. Greig, M. Hodgetts, and J. Gilby, "*An EM / E-MRF algorithm for adaptive model based tracking in extremely poor visibility*," vol. 26, pp. 480–495, 2008.

[50] J. Zhang, F. Shi, J. Wang, and Y. Liu, "*3D Motion Segmentation from Straight-Line*," pp. 85–94, 2007.

[51] X. Llad, "*Euclidean Reconstruction of Deformable Structure Using a Perspective Camera with Varying Intrinsic Parameters o*," pp. 3–6.

[52] A. Goh and R. Vidal, "*Segmenting motions of different types by unsupervised manifold clustering*," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, 2007, pp. 1–6.

[53] B. Shoushtarian, "*A practical approach to real-time dynamic background generation based on a temporal median filter*," *J. Sci. Islam. Repub. Iran*, vol. 14, no. 4, pp. 351–362, 2003.

[54] P. W. Power and J. A. Schoonees, "*Understanding background mixture models for foreground segmentation*," in *Proceedings image and vision computing New Zealand*, 2002, vol. 2002.

[55] I. Haritaoglu, D. Harwood, and L. S. Davis, "*W/sup 4: real-time surveillance of people and their activities*," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 809–830, 2000.

[56] R. Cucchiara, C. Grana, A. Prati, A. Member, and R. Vezzani, "*Probabilistic posture classification for human-behavior analysis*," *IEEE Trans. Syst. man, Cybern. A Syst. Humans*, vol. 35, no. 1, pp. 42–54, 2005.

[57] W. Hu, T. Tan, L. Wang, and S. Maybank, "*A survey on visual surveillance of object motion and behaviors*," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 34, no. 3, pp. 334–352, 2004.

[58] D. Choi and B. Van Roy, *A Generalized Kalman Filter for Fixed Point Approximation and Efficient Temporal-Difference Learning*. 2006.

[59] K. Nummiaro, E. Koller-meier, and L. Van Gool, "*An adaptive color-based particle filter*," vol. 21, pp. 99–110, 2003.

[60] E. Zhang, F. Chen, and W. Zhang, "*A novel particle filter based background subtraction method*," in *Computational intelligence and security, 2006 international conference on*, 2006, vol. 2, pp. 1837–1840.

[61] P. O. Donovan and P. O'Donovan, "*Optical flow: Techniques and applications*," *Int. J. Comput. Vis.*, pp. 1–26, 2005.

[62] K. Liu, S. S. Lienkamp, A. Shindo, J. B. Wallingford, G. Walz, and O. Ronneberger, "*Optical flow guided cell segmentation and tracking in developing tissue*," in *Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on*, 2014, pp. 298–301.

[63] T.-C. Huang, C.-K. Chang, C.-H. Liao, and Y.-J. Ho, "*Quantification of blood flow in internal cerebral artery by optical flow method on digital subtraction angiography in comparison with time-of-flight magnetic resonance angiography*," *PLoS One*, vol. 8, no. 1, p. e54678, 2013.

[64] M. Xavier, A. Lalande, P. M. Walker, F. Brunotte, and L. Legrand, "*An adapted optical flow algorithm for robust quantification of cardiac wall motion from standard cine-mr examinations*," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 5, pp. 859–868, 2012.

[65] N. Hata *et al.*, "*Three-dimensional optical flow method for measurement of volumetric brain deformation from intraoperative MR images*," *J. Comput. Assist. Tomogr.*, vol. 24, no. 4, pp. 531–538, 2000.

[66] A. Geiger, P. Lenz, and R. Urtasun, "*Are we ready for autonomous driving? the kitti vision benchmark suite*," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012, pp. 3354–3361.

[67] Z. Sun, G. Bebis, and R. Miller, "*On-road vehicle detection: A review*," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 694–711, 2006.

[68] G. Piriou, P. Bouthemy, and J.-F. Yao, "*Recognition of dynamic video contents with global probabilistic models of visual motion*," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3417–3430, 2006.

[69] R. Gal, N. Kiryati, and N. Sochen, "*Progress in the restoration of image sequences degraded by atmospheric turbulence*," *Pattern Recognit. Lett.*, vol. 48, pp. 8–14, 2014.

[70] M. Werlberger, T. Pock, M. Unger, and H. Bischof, "*Optical flow guided TV-L1 video interpolation and restoration*," in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2011, pp. 273–286.

[71] S. S. Kumar and M. John, "*Human activity recognition using optical flow based feature set,*" in *Security Technology (ICCST), 2016 IEEE International Carnahan Conference on*, 2016, pp. 1–5.

[72] M. Paul, S. Haque, and S. Chakraborty, "*Human detection in surveillance videos and its applications-a review,*" *EURASIP J. Adv. ...*, pp. 1–16, 2013.

[73] I. Journal and I. Theories, "*HIGH LEVEL COLOR SIMILARITY RETRIEVAL Peter L . Stanchev , David Green Jr ., Boyan Dimitrov,*" vol. 10, pp. 283–287.

[74] D. Zhang, M. Islam, and G. Lu, "*A review on automatic image annotation techniques,*" *Pattern Recognit.*, vol. 45, no. 1, pp. 346–362, 2012.

[75] S. L. Feng, R. Manmatha, and V. Lavrenko, "*Multiple bernoulli relevance models for image and video annotation,*" in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004, vol. 2, pp. II--II.

[76] J. Fan, Y. Gao, and H. Luo, "*Automatic Image Annotation by Using Concept-Sensitive Salient Objects for Image Content Representation,*" pp. 361–368.

[77] D. P. Tian, "*A review on image feature extraction and representation techniques,*" *Int. J. Multimed. Ubiquitous Eng.*, vol. 8, no. 4, pp. 385–395, 2013.

[78] D. Zhang and G. Lu, "*Review of shape representation and description techniques,*" vol. 37, pp. 1–19, 2004.

[79] A. Toshev and C. Szegedy, "*DeepPose : Human Pose Estimation via Deep Neural Networks.*"

[80] M. Andriluka, S. Roth, and B. Schiele, "*Pictorial structures revisited: People detection and articulated pose estimation,*" *2009 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work. CVPR Work. 2009*, vol. 2009 IEEE, pp. 1014–1021, 2009.

[81] M. Eichner and V. Ferrari, "*Better appearance models for pictorial structures,*" *Procedings Br. Mach. Vis. Conf. 2009*, p. 3.1-3.11, 2009.

[82] Y. Yang and D. Ramanan, "*Articulated pose estimation with flexible mixtures-of-parts*," *Comput. Vis. Pattern …*, 2011.

[83] J. Tompson and G. W. Taylor, "*Learning Human Pose Estimation Features with Convolutional Networks*," pp. 1–11.

[84] A. Yilmaz, O. Javed, and M. Shah, "*Object tracking A Survey*," *ACM Comput. Surv.*, vol. 38, no. 4, p. 13–es, 2006.

[85] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. Van Den Hengel, "*A survey of appearance models in visual object tracking*," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 4, p. 58, 2013.

[86] J. Ho, K.-C. Lee, M.-H. Yang, and D. Kriegman, "*Visual tracking using learned linear subspaces*," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004, vol. 1, pp. I--I.

[87] C. Yang, R. Duraiswami, and L. Davis, "*Efficient mean-shift tracking via a new similarity measure*," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE computer society conference on*, 2005, vol. 1, pp. 176–183.

[88] H. Zhou, Y. Yuan, and C. Shi, "*Object tracking using SIFT features and mean shift*," *Comput. Vis. Image Underst.*, vol. 113, no. 3, pp. 345–352, 2009.

[89] S.-W. Ha, "*Multiple Objects Tracking with Location Matching and Adaptive Windowing Based on SIFT Algorithm*," *Compusoft*, vol. 2, no. 12, 2013.

[90] Y. Zhu, B. Dariush, and K. Fujimura, "*Kinematic self retargeting: A framework for human pose estimation*," *Comput. Vis. image Underst.*, vol. 114, no. 12, pp. 1362–1375, 2010.

[91] D. Vlasic, I. Baran, W. Matusik, and J. Popović, "*Articulated mesh animation from multi-view silhouettes*," in *ACM Transactions on Graphics (TOG)*, 2008, vol. 27, no. 3, p. 97.

[92] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers, "*An improved algorithm for*

tv-l 1 optical flow," in *Statistical and geometrical approaches to visual motion analysis*, Springer, 2009, pp. 23–45.

[93] C. Zach, T. Pock, and H. Bischof, "*A duality based approach for realtime TV-L 1 optical flow*," in *Joint Pattern Recognition Symposium*, 2007, pp. 214–223.

[94] R. Girisha and S. Murali, "*Tracking humans using novel optical flow algorithm for surveillance videos*," in *Proceedings of the Fourth Annual ACM Bangalore Conference*, 2011, p. 7.

[95] S. Denman, V. Chandran, and S. Sridharan, "*An Adaptive Optical Flow Technique for Person Tracking Systems*," vol. 28, pp. 1232–1239, 2007.

[96] J. K. Aggarwal and M. S. Ryoo, "*Human activity analysis: A review*," *ACM Comput. Surv.*, vol. 43, no. 3, p. 16, 2011.

[97] R. Poppe, "*A survey on vision-based human action recognition*," *Image Vis. Comput.*, vol. 28, no. 6, pp. 976–990, 2010.

[98] X. Yang and Y. L. Tian, "*EigenJoints-based action recognition using Naive-Bayes-Nearest-Neighbor*," *2012 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 14–19, 2012.

[99] Z. F. Huang and G. Mori, "*SFU at TRECVid 2010 : Surveillance Event Detection*," *Trecvid 2010*, pp. 1–6, 2010.

[100] K. Tang, L. Fei-Fei, and D. Koller, "*Learning latent temporal structure for complex event detection*," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1250–1257.

[101] G. Yao, T. Lei, and J. Zhong, "*A review of Convolutional-Neural-Network-based action recognition*," *Pattern Recognit. Lett.*, vol. 0, pp. 1–9, 2018.

[102] K. Yokoi, T. Watanabe, and S. Ito, "*Toshiba at TRECVID 2009 : Surveillance Event Detection Task*," 2009.

[103] M. Dikmen *et al.*, "*Surveillance Event Detection*," in *TRECVID Video Evaluation*

Workshop, 2009.

[104] M. Han, W. Xu, H. Tao, and Y. Gong, "*An algorithm for multiple object trajectory tracking*," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004, vol. 1, no. C, pp. I--I.

[105] R. H. Baxter, N. M. Robertson, and D. M. Lane, "*Real-time event recognition from video via a ' bag-of-activities ,"*" 2009.

[106] D. Ward, "*Frame Rate Guide for Video Surveillance*," 2014. [Online]. Available: https://ipvm.com/reports/frame-rate-surveillance-guide.

[107] C. Feichtenhofer, A. Pinz, and A. Zisserman, "*Convolutional Two-Stream Network Fusion for Video Action Recognition*," no. i, 2016.

[108] D. Tran, S. Member, J. Yuan, and D. Forsyth, "*Video Event Detection : from Subvolume Localization to Spatio-Temporal Path Search*," 2014.

DECLARATION

I, the undersigned, declare that this research is my original work and has not been presented for degree in any other university, and that all sources of materials used for the research have been acknowledged.

Declared by:

Name: **Daniel Sahle Chemere**

Signature: _____

Date: _____

Confirmed by advisor:

Name: **Yaregal Assabie (PhD)**

Signature: _____

Date: _____

Place and date of submission: Addis Ababa University, October 5, 2018.