Anomaly Detection in Real Time CCTV Streams

D.M.B. Kularathne

179330X

Degree of Master of Science

Department of Computer Science Engineering

University of Moratuwa
Sri Lanka

July 9, 2020

Declaration

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another per-son except where the acknowledgement is made in the text.
Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: ..... Date: .....

Name: D. M. B. Kularathne

The above candidate has carried out research for the Masters/MPhil/PhD thesis/Dissertation under my supervision.

Signature of the supervisor: ..... Date: .....

Name: Dr. Charith Chitraranjan

Abstract

Anomaly detection in video data has been a challenge always. After the introduction of many state-of-art designs, this still poses a challenge as those systems may fail to work in all types of environments. Even though many supervised methods claimed to have some good results in this domain, supervised systems may not be suitable for all the contexts such as in an open area, any type of anomaly can occur and it can be very difficult to train a system in a supervised manner to identify an unanticipated anomaly. On the other hand, it would be difficult for the user to annotate data each time when they change the context under surveillance for the device. Thus the ultimate solution should be an unsupervised solution with a appreciable accuracy. Recently deep learning techniques have emerged in many areas of computer science based solutions and so it is involved for anomaly detection tasks also. In this research, deep learning techniques are involved to solve the problem of video stream based anomaly detection of crowds.

Acknowledgements

I would like to express my gratitude for my supervisor Dr. Charith Chitranjan for the guidance and support extended since beginning. His invaluable advice greatly helped me to drive this research work on the right track. I further would like to thank my colleagues from the MSc batch who shared their knowledge and insights with me.

I am certainly in debt to my parents who guided me throughout the journey of life. I wish to give my heartiest thanks to my loving wife who stood beside me, encouraged me and supported me throughout my work and this would not have been possible without her immense support and love. Finally, I would like to extend my gratitude to all the colleagues at Synopsys who helped me in various ways to fulfill this task.

Contents

List of Figures

List of Tables

Abbreviations

AAE Adversarial Autoencoder. 24, 36, 55

ADAE Adversarial Dual Autoencoder. 29

Adam Adaptive Moment Estimation. 39

AE Autoencoder. 16, 18, 19, 21, 36

AUC Area Under Curve. ix, 32, 46, 47, 49, 52, 59, 66, 67

CAE Convolutional Autoencoder. 18–20, 31

CNN Convolutional Neural Network. 18, 19, 25, 28, 30

ConvLSTM Convolutional Long-Short-Term-Memory. v, 26

EER Equal Error Rate. ix, 32, 46, 47, 49, 52, 59, 66, 67

GAN Generative Adversarial Network. v, vii, 17, 29–31, 35

GIGO Garbage In Garbage Out. 4

GMM Gaussian Mixture Model. 13

HDP Hierarchical Dirichlet Process. 12

HMM Hidden Markov Model. 11, 12

LDA GLatent Dirichlet Allocation. 12

LSTM Long-Short-Term-Memory. v, 17, 25, 26, 28, 32, 33, 36, 37

MAE Mean Absolute Error. 24, 36

MSE Mean Squared Error. 24, 25, 31, 36, 39

PCA Principal Component Analysis. v, 16–20, 28

RBM Restricted Boltzmann Machine. 24

ReLU Rectified Linear Unit. 18

RL Representation Learning. v, 16, 17

RNN Recurrent Neural Network. 25

ROC Receiver Operating Characteristic. 46

SFA Slow Feature Analysis. v, 26–28

STSAE Spatio-Temporal Stacked frame Auto Encoder. 20

VAE Variational Autoencoder. 17, 36

List of Appendices

Chapter 1

Introduction

The world is moving towards making the next generations safety mechanisms for their communities due to the lack of security in public places. Thus every country recently has invested largely in those security related domains such as mapping of incidents throughout the world, social graph analysis, and most importantly surveillance. The governmental agencies are equipping their security forces with tools and skills to quickly react for any sudden event that could occur in public places so that they can minimize the damage being done. But the information is sometimes delayed and thus the damage becomes large due to the lack of incorporation of real time information sources. For this reason, investing in real time surveillance systems has become major concerns nowadays where capturing of anomalous events in the real time has gained much focus than ever in the history. One sub section of the real time surveillance domain is detection of anomalies in crowded scenes. The importance of detection of anomalies in crowds is mainly contributing to the social security monitoring activities. If an abnormal event like a street fight or some other unpredictable event happens these systems can have a flag in their security cameras or in an extreme condition, they also can alert the relevant authorities based on the severity of the anomaly.

Human activity recognition has always been a challenge throughout many years and the improvements of the subject has lead the focus onto real time activity recognition. Crowded scenes analysis is one of the sub categories that fall under the vast scope of human activity analysis. These crowded scene analysis methodologies are used in real time environments and the major area that lies is the detection of the abnormal events or in other words the anomalies in the scene. In the past few years a massive amount of research has been performed in the domain of human pose, activity and anomaly classification and especially in terms of anomaly detection. The real value of such mechanisms is lying in the domain of real time automated surveillance for the security sector. Thus the main focus of this research is driven with the same idea that improving the detection of abnormalities for the surveillance with some automated mechanism will benefit immensely when it comes to the vast amount of dynamic varieties of contexts in the real world scenarios. For example, the detection of anomalies in a busy market area would be entirely different from a subway area where there are periodic movements of people in a specific pattern and less movements during other times. So with the diversity of these environmental contexts, the importance of an unsupervised and accurate surveillance monitoring system has emerged. Another driving factor is the overabundance of the surveillance data that are available versus the manpower that are available for processing them

manually is of shortage.

Anomaly detection narrows down to two classes in terms of what to detect. The model can either classify two classes or one class or in other words it has to learn both normality features and abnormality features or it can learn either of those only in order to distinguish one from the other. The most popular and practical method is to detect one class. Anticipation of the types of the abnormalities by learning the abnormality feature was one of the older approaches which failed due to various reasons. The major reason is that the fact that the anomalies are rare and the types are unimaginable such that one cannot comprehensively train a system effectively to detect all kinds of anomalies. So the state of art systems tend to use the reverse of this approach which is detection of the anomalies via the prior knowledge of the normality features of the scene. When the system can predict the what will happen in the near future, any event that significantly deviates from the predicted metrics would be considered as an anomaly with respect to the context.

1.1 Anomaly Detection

Anomaly detection or novelty detection refers to the problem of detecting patterns in data that fall away from the expected behavior. The anomalies are also referred to as outliers in the literature. There are many use cases of anomaly detection. Fraud detection for credit cards, intrusion detections, anomalies in distributed systems, fault detection in critical systems, and most commonly in military and security systems in public areas.

1.1.1 Definition of Anomalies

Anomalies can be stated as the pattern that do not conform to a well-defined notion of normal behavior. Anomaly detection is sometimes wrongly identified as noise removal. But noise removal has its own definition that can be clearly distinguished from anomaly detection. Noise is nothing but some unwanted piece of data which in fact acts as a hindrance for the real analysis task but on the other hand anomalies often become the whole purpose of analysis.

1.1.2 Challenges in Anomaly Detection

Even though it is very straightforward to state the process as a formal definition, the process is a very challenging task due to several reasons. One main challenge is that there is no generic measure to interpret the normality due to the domain specific variance.Thus a methodology in one domain is not as straightforward as it seems to be applied in a different domain. It is generally not straightforward to encompass an area that will contain all the normal behavior as due to the difficulty of capturing all the possible normal behaviors. The boundary between the normal and abnormal behavior is often blurred and some anomalies which lie close to the normal behavior are often hard to distinguish, and vice-versa. Another challenge is the lack of availability of labeled data. For example, in a video based anomaly detection approach, the model may need sufficient amount of anomalous events in a video stream to train effectively. But there is always a shortage of sufficient amount of anomalous events present in the video frames. So the analysts will have to use some data augmentation methodologies in order to feed sufficient amount of data.

1.1.3 Aspects of Anomaly Detection Problem

This section elaborates on the different aspects of the anomaly detection problem. The problem is viewed as a broach domain and different aspects of the problem space is discussed briefly.

1.1.3.1 Data As Input

One of the most important aspects is the nature of the input data. In all kinds of analysis approaches the GIGO(Garbage In Garbage Out) principle is the key rule applicable. This essentially means that if the input is garbage, the output is naturally becoming garbage. Input is nothing but a collection of data instances where a small portion of that will be noise or in other words garbage. So as an analyst, it is essential to get rid of these garbage before starting the analysis.

1.1.3.2 Type of Anomaly

Another important aspect is the type of anomaly. The type basically can be categorized into following three categories.

- Point Anomalies

- Contextual Anomalies
- Collective Anomalies

Point Anomalies
If an individual point is identified as an anomaly, compared to rest of the data, then that point is identified as point anomaly.

Contextual Anomalies
If a data point is considered to be anomalous in a given specific context, and not otherwise, then such anomalies are called contextual anomalies as they consider the context in which the specific behavior becomes an anomaly.

Collective Anomalies
In this type of anomaly, the requirement is that a set of data points being anomalous with respect to the entire data set. In this case an individual anomalous point taken alone may not be an anomaly but rather taken as a collection, they may form an anomalous behavior altogether.

1.1.3.3 Availability of Labeled Data

The data labels are used to denote whether a particular data point is anomalous or not. But the most exhaustive task regarding the labeling is obtaining data to be labeled for all types of anomalous event which is often very expensive in nature. The labeling process is often done by a domain expert who decides on the label and in this process the human expert will have to go through each and every data point and label them accordingly which is a cumbersome process. Obtaining labeled data for all types of anomalous data is generally more difficult than getting a set of labeled points representing the normal behavior. Since the anomalous behavior is often dynamic in nature, many new types of anomalous data might appear where such labeled data is not present at the moment. For example, in a video stream, if we have all types of anomalous behaviors observed so far labeled properly, there would still be a new behavior of a person who is acting differently than all other people observed so far. Based on the extent to which the labels are available, anomaly detection approaches can operate in one of the following three modes:

- Supervised anomaly detection

- Semi-supervised anomaly detection
- Unsupervised anomaly detection

Supervised anomaly detection
In this detection method the data labels are expected to be present in both the classes anomalous and normal. Often the methodology used in such cases is training a model in order to capture the normal and abnormal class in a predictive manner. Thus if a new data point is encountered, the model is used to classify the class that it belongs to. In this methodology, there are two basic issue that may arise. The main issue is the amount of anomalous data is of shortage. In order to overcome this issue, we can use data augmentation methodologies. This problem can also be interpreted as the problem of class distribution being imbalanced. The other issue, which is an issue that is tightly bound with the latter problem, is the inability to obtain accurately for the anomalous class.

Semi-supervised anomaly detection
This is the most popular way of making a model for anomaly detection. In this method, the basic assumption is that assuming the labeled data exists only for the normal class. This method addresses the two main issues faced with the supervised methods and thus this method is widely applied in the practice. In this approach typically the anomalous data is defined as whatever the data points that deviate from the assumed normal model. It is also important to note that there are other methods as well in which the basic idea is to have labeled data only for the anomaly class. These techniques are not commonly used due to the fact that it is not often possible to obtain a data set that covers all types of anomalous behaviors as the space of anomalous probabilities are not comprehensively perceivable and often ambiguous.

Unsupervised anomaly detection
This is the scenario where the training data is not required thus the applicability is high. The methods incorporate an implicit assumption that normal instances are frequent and the anomalies comprise only a smaller portion. This leads to a high false alarm rate if the assumption does not comply with the data. In order to overcome this issue, the model trained using a semi-supervised model can be used against the data and that will provide a robust methodology that will handle the few anomalies.

1.2 Abnormal Human Behavior Recognition

Due to the increased global security concerns, intelligent vision based solutions has gained more focus in the modern era. The most attractive research area is monitoring human behavior and patterns in surveillance footage. The idea is to learn, model, detect, or recognize interesting events that may be defined as suspicious events related to human behaviors in crowded scenes.

The task is not a straight forward due to a number of reasons. The following section describes the difficulties that can be identified when it comes to video based abnormal human behavior modeling.

1.2.1 Challenges of modeling the human behavior in videos

One of the main challenges is that the video stream itself being high dimensional, the stream cannot directly be fed into a classifier. The reason is that they contain much redundant information and thus cause high computational complexity. So the video data has to be represented in a manner that can be efficiently processed and be accurate on the task given. The key to a successful model is choosing a suitable representation which is also the most challenging task of all.

- **Same class of action having a variety of patterns**
  One class of action can vary immensely that can be very hard to represent in a manner that is generic enough to capture all the small variations in the same class but distinctive enough to distinguish between other classes. For example, a walking person might carry a bag or might interact with another person, but the behavior should still be captured under the walking class. But if a person starts running, the model should be able to distinguish the running person from a walking person.

- **Noise**
  In a real world scenario, the scenes would contain a lot of noise due to various reasons. The reason may be a background change or a change in the illumination. One of the key expectations of a robust system is that the ability to work under various environments regardless of the expected level of noise.

- **The Context**
  The contextual information is also a crucial factor that affects the accuracy of the model. If the scene to be analyzed is a crowded scene, the behavioral

model need to take account of the gathered gestures of humans. This challenge is not present in an isolated environment where there are only a few people walking in the area. One example is a factory environment where there are only a few workers walking and performing various tasks. But in a crowded pedestrian environment, the modeling can be much challenging due to the high density of the people in one location.

- **The Illumination based on the time of day**
  This is when the illumination varies along with the time of the day. In the day time the detection would be much easier than at night. During the night time a night vision camera can be utilized but the image preprocessing may have to be altered based on the illumination.

1.3 Problem

The main research problem trying to address via this research is that to develop a human abnormality detection system that works well on crowded scenarios. Currently there are many systems that are capable of capturing anomalies in crowds, but those systems are developed based on conventional methods. Along with the recent uprising of the deep learning domain, such systems could be developed to be more accurate and adaptable to various environments. Many of those conventional systems needed to be highly supervised during the training period as the anomalies can differ from situation to situation. But with novel deep learning approaches those systems can be improved to be semi-supervised or unsupervised, which would be a great leap above many barriers that were there in this domain. Main issue of using a supervised system is that the user will have to manually annotate the data and feed into the system. In case if the user decides to change the location of the device and aim it to a different context, the system will have to be re-trained with a new set of annotated data, which is very cumbersome as the user will have to manually annotate data each time the context is changed. If the system requires also the anomalous data to be annotated and fed apart from the normal data, there could be a difficulty in providing anomalous data due to the fact that the anomalous data is not easy to gain. In comparison to normal data, anomalies are rare scenarios and may not be in a sufficient amount to be fed to a learning algorithm. On the other hand the anomalies cannot be limited to a certain set of categories due to the diverse nature of the events. Anomalies are mostly unintended and unanticipated. Hence

the system should be able to identify anything outside the scope of a normal event. Another issue of supervised learning is that these systems are ultimately expected to be manufactured in a production level and if the users have to train their systems in a supervised manner, then that would require the users to be equipped with a data science knowledge, which is very unlikely the case of a real world scenario. The users need to be able to plug and play such a device with a minimum level of configurations that can be provide by an average user. Hence, the only option left is make the system unsupervised and let the users only be aware that there is a training period before actual usage. Due to this requirement, deep learning methodologies become very useful as they can be trained as a black box with a minimum level of supervision. There are currently some deep learning based systems developed for the problem, but those systems lack adaptability to the environment and also since those are commercial systems, they are not actively used in 3rd worlds countries due to unaffordability.

Since the deep learning approaches unarguably deliver better accuracies and atop amongst other systems in terms of adaptability and robustness with the proper amount of data given, this research is focusing only on deep learning approaches. Nevertheless, for the sake of comprehensiveness, other conventional approaches are also noted under the literature review.

1.4 Objectives

The main objective of this research is to create a tool that can identify anomalies in a crowd situation in the real time. The general objective of the research can be stated as below. To create a tool that can accurately flag anomalous events in crowd scenarios in a real time video frame. The anomalous event can vary from situation to situation, and the tool should still be able to successfully distinguish those anomalous events regardless of the context.

The following items listed are the important characteristics of the system.

- **The system should be adaptable**
  The system should be adaptable, ie. this system should simply be able to be adapted to any environment with some fully unsupervised training sessions.

- **Should not be constricted to a certain type of anomalies**
  With the proper amount of training data given, the system should distin-

guish the anomalous events of a vast diversity and should not be constricted to a certain type of anomalies.

- **Should be able to identify context based anomalies**
  The anomalies can differ from context to context. For example, on a pedestrian pathway, a person running would be treated as an anomaly. But on a jogging pathway, a person running is a normal scene. The system should be able to find anomalies based on the context.It should be able to define what is anomalous and what is not by comparing with the usual context.

- **Should have a good sensitivity**
  The system should identify the true positives correctly and it is tolerable to have a certain amount of false positives. The false negatives should be avoided as much as possible.

  Another objective of this research is to explore how to minimize the data requirement with the introduction of a suitable architecture. Also with that architecture, the system is expected to be more robust and accurate. Exploration of different deep neural architectures that can cater these requirements hence stands amongst the main objectives of this research.

  This tool could be used for crowd observation security purposes. In case of any anomalous event, a flag would be inserted along with the time frame, and later could be allowed to be viewed by any interested party.

1.4.1 Specific Objectives

- To develop a deep neural architecture that can cater for the expected performance measures (low false negative rate/ higher sensitivity)
- Make the system adaptable to crowded environments
- Improve the flexibility of the tool
- Improve the sensitivity(reduce the amount of true negatives) of the tool
- Improve the reliability of the tool
- Future extension: Draw/Mark the area in the video frame that is identified as anomalous

1.5 Prior Work

1.5.1 Models and Algorithms for Abnormality Detection in Video Streams (Conventional Models)

This section discusses about the conventional methods of anomaly detection that were there before the massive up-wave of the deep learning usage that was made possible by the recent advancements of hardware like GPUs. Those methods basically consist of clustering and classification methodologies that were mostly supervised. Unsupervised methods were not very easy to model as those conventional methods mostly required a reference target to be matched against.
The method of evaluating spatial information was to consider local patches and run the training algorithms in multi-scale mode. The way to model temporal dependency was to incorporate HMMs. These methods needed careful feature engineering and mostly a knowledge about the anomaly types that can occur. These models were not robust towards the new types of anomalies. On the other hand providing at least those few anomaly classes was also a challenge due to lack of anomalous events captured.
Since the features had to be manually engineered, these models did not have the capability to go up to high accuracy values where in deep learning models the same is a possibility.

This section briefly introduces a few most commonly used conventional models and their strengths and weaknesses.

1.5.1.1 Dynamic Bayesian Network

HMM is one of the most popular methods for behavior modeling and this fact is well utilized by the domain of anomaly detection. It has gained this much popularity in this domain probably due to the inherent temporal dependence achieved by it in its own nature. HMM models, unlike other models, are capable of handling inherently dynamic behavior patterns applied in the domain of anomaly detection.HMM is basically a set of nodes connected according to a time series structure and the nodes are connected via transition links. Every node has a corresponding state that is hidden hence the name Hidden Markov Model. An observation is made at each state and this observation corresponds to a set of state probabilities. Most commonly an HMMs are represented by two matrices that represent the probable states and the probability of their observations. Those

two matrices are namely the emission matrix and the state transition matrix.

There have been various prior studies in HMM modeling which primarily differ in three main aspects. The formation of the model, meanings of observations, states assigned to nodes are them. Nodes can used to represent any concept like positions, movement metrics, or even postures which are crowd behavior or could be also some local behaviors like any type of individual behaviors like walking, standing & etc. Observations would represent some crowd activity level or some quantitative measurement.

There have been two drawbacks that were under the focus as per mentioned by the authors of [1]. These were mentioned respective to their area of study. But it could be an important note for anyone studying in any application of anomaly detection. The first point that they mention is that difficulty of foreseeing the trends of anomalies that may not be visible presently but in the future. This inability may lead the system to fail in detecting sudden changes of the environment. Decisions are made considering a particular context and this would lead to considerably higher false alarm rates. This is the second key point that they had listed. To overcome the aforementioned difficulties, they had developed a system that integrates Fuzzy Logic with HMM. The techniques that they employed were two fold. One is considering the whole training data set as normal data. The other is integrating the some amount of anomalous data to the training data. If the whole training data set is considered as normal data, there is a necessity of using a threshold value to bound the normal data and identify the abnormalities. In the case of incorporating abnormal data in the training set, they had to use two hidden states.

1.5.1.2 Bayesian Topic Models

Many efforts have been invested in Bayesian Topic Models to that were able to evaluate the regularity if local events(word) while looking into interactions(topic) between them [2],[3].These were only run in batch mode[4]. These approaches did not require them to be fed them explicitly of any spatial or temporal dependencies between local events. There are two models, namely LDA and HDP. LDA stands for Latent Dirichlet Allocation and HDP stands for Hierarchical Dirichlet Process. These are basically hierarchical Bayesian models and they were utilized for processing linguistics[2]. Authors in [2] are proposing a method to improve the models LDA and HDP using hierarchical Bayesian models. The intention was to model the interactions unsupervised. They were able to provide a probabilis-

tic explanation to surveillance tasks such as anomaly detection and clustering in video sequences. Over-fitting issue would be avoided due to the availability of sufficient parameters coming from a hierarchical model as the data is hierarchical. However this was not able to model global behavior patterns and failed in modeling complex behaviors[3]. The reason was the fact that this approach only focuses on the local motion features. This was also not able to model correlations between moving and fixed objects due to the same reason.

1.5.1.3 Clustering

Clustering can be done without labeling the data or in other words unsupervised. In addition to fully unsupervised clustering, in [5] and [6], semi-supervised clustering is also explored. Clustering process is very expensive in terms of speed and resource consumption for computations. This had lead for this method to be less used for the abnormal detection tasks especially when it comes to complex unsupervised learning tasks with many classes involved even though it is smooth and fast after the clusters are detected. The clustering process becomes the critical factor of deciding the performance of the anomaly detection algorithm. If the process leads to bad clusters, the same would lead to bad detection [7]. The k-means is one of the broadly used algorithms to cluster features. Some researchers have worked on further advanced improvements that have made them overcome the limitations of k-means when implemented for behavior clustering like k-medoids [8], radius-based clustering [9], and ant-based clustering [10].

In general applications, model-based clustering algorithms unlike the k-means do not expect a predetermined number to be fed as the total number of clusters expected. But without having a knowledge about the data and its distribution beforehand, these methods might be a bit difficult to develop. Gaussian mixture model (GMM) [11] is an excellent method in which the number of clusters is derived from a Gaussian distribution [9]. However, in the researches [12],[13],[14],[15],[16],[17], GMM is used to detect anomalies in automated surveillance streams.

1.5.1.4 Decision Trees

A decision tree's structure is nothing but a structure of nodes formed in a cascaded manner. The node connections are formed in a structure of a tree and each node belonging to a layer that defines a certain depth of the tree in which each node is connected to upper and lower layer in a manner where upper layer is parent

and only one parent is connected.But there can be any number of children nodes connected from the lower layer. The interpretation capability is one of the main highlights when it comes to using a decision tree. Decisions are represented by each node and connections are representing states and their probabilities. Decision tree is one of the most common techniques for representing classifiers. A decision tree can be either regression or classification tree depending on the nature of the target variables For continuous data, regression tree is used and for discrete data, a classification tree is formed. Nodes represent decisions and branches represent the transition probability of entering into states.
In the research carried out in [18], the authors introduced a new method for anomaly detection using an N-array tree classifier. In this method, the classifier's tree is formed into different layers in which each layer represents a certain period of time. A supervised way was used to learn the probabilities of the tree links from both regular and irregular training data instances. After a number of training iterations, a formerly unseen behaviour is learned. If a higher probability is shown for entering a particular state, its probability of entering is evaluated for each connecting state. The higher the probability the higher the possibility of entering an anomaly state.

The conventional methods were the most successful till the time the deep learning took over with the uprising of the vast amount of data available alongside improved processing power which is specialized in parallel processing huge chunks of data. Under the literature review, more into the deep learning related work will be discussed.

Chapter 2

Literature Review

2.1 Anomaly Detection using Deep Learning

In the following section, recent deep learning models for anomaly detection tasks that were reviewed and compared with their pluses and minuses. The recent focus in this areas has been towards the generative models but none of the prevailing models have yet been tried with sequence generation of crowd data but the 2 dimensional MNIST data. The most recent researches for crowd anomaly detection have been performed by [19], [20], [21], and [22].

2.1.1 Representation Learning(RL)

The representation learning as per the definition is as following. Building a parameterized model f_x ,such that $f_x : X \rightarrow Z$. This can either be input domain to a lower dimensional space or to the input domain itself. Z is generally invariant to the local changes of the input. In this case modeling expect prior information such as transformations in the normal sample points. In the context of this research, this is modeling of the spatio-temporal regularity, trajectory or local relative motion and temporal correlation of the structure.For anomaly detection tasks in video surveillance, the most famous type of modeling technique is the representation learning. This very domain can be categorized under three categories that can be used for anomaly detection purpose.

2.1.1.1 RL For Reconstruction

The idea here is to reconstruct a given image by using a generative model training method. The successful recreations are understood as non-anomalous frame. The more it deviates from being a successful recreation, the more anomalous it is. Thus any frame representation that is poorly reconstructed are considered as to be an anomaly. Deep learning approaches like (PCA), and AE that can be used to model how to compress temporal and spatial information or in other words the, image and the flow of the objects by modeling the normal behavior in surveillance videos, can be categorized under this section.

2.1.1.2 RL for Predictive Modeling

The video frames are viewed time series or in other words temporal data. The models are supposed to observe the past video frames and predict the current video frame or any representation that can be generated from the cur-

rent video frame. The basic idea here is to construct a basic conditional model $P(x_t|x_{t-1}, x_{t-2}, x_{t-3}....x_{t-p})$. Auto-Regressive models and Convolutional LSTMs generally come under this category.

2.1.1.3 RL for Generative Models

For the supervised learning setup, $(X_i, y_i) \in R \times \{C_j\}_{j=1}^{K}$ where i is the index number of the samples $i = 1 : K$ in the data set, generative models estimate the class conditional posterior probability distribution $P(X \mid y)$. This can be difficult in case of a higher d, the dimentionality. The spatio-temporal video streams can thus be a challenging input for these models. In order to model the likelihood of normal video samples in an end to end deep learning structure, Generative Adversarial Networks (GAN), Variational Autoencoders (VAE), and Adversarial Auto-Encoders (AAE) can be used.

2.1.2 Reconstruction Models

Let's consider an input training video that is represented as below. $X \in R^{(N \times d)}$ where N is the number of frames, and d is pixels per each frame as given by $d = r \times c$. The degree of dimensions of each vector is represented by this . The main goal of the methods under this section is to reduce the expected reconstruction error. Convolutional Auto-Encoder (ConvAE), Contractive Auto-Encoders (CtractAE), and Principal Component Analysis (PCA) are described in detail under this section. Their structures will be described in the purpose of reconstruction and the reduction of dimensions.

2.1.2.1 Principal Coponent Analysis(PCA)

In PCA it basically attempts to find the direction of the maximul variance in the training data which in this case is nothing but the video frames. The main goal of representing videos is that modeling information that is contained in the form of spatial and temporal dimensions which in turn would become the principal components of a vector representing a video at a given time step t.

$$\min_{W^T W = I} \|X - (XW)W^T\|_F^2 = \|X - \tilde{X}\|_F^2 \tag{2.1}$$

Where $W \in R^{d \times k}$ is a matrix that has a lower number of columns or in other words lower number of components than X and XW represents the projection into lower dimensional subspace. This dimensionality reduction can be utilized

in identifying the novelty behavior. The output which are not properly reconstructed or reconstructed with an error above a predefined threshold are identified as anomalies. Mahalanobis distance is used to as the anomaly score.

2.1.2.2 Auto-Encoders

Autoencoder is an alternative to the PCA with some additional functionality that essentially can be used in the same way as PCA to reduce the dimensionality. But one advantage that the autoencoders possess is that the flexibility of using non linear activation functions that will enable the AEs to find a different subspace than the PCA. Otherwise AEs would be equivalent to the result of the PCA. A single layer auto encoder with linear transfer function can be stated as similar or closely equivalent to PCA, where closely means that the W parameters calculated by AE and PCA may not be similar; but the latent space spanned by the W s will be similar.

The autoencoder functionality is as below. It takes a input of $z \in R^d$ and maps the input to the latent space representation $z \in R^k$ with dimensionality reduced ($d > k$). This was done using a function $z = \sigma(Wx + b)$

The non-linearity of AutoencodersAE is gained by using a non-lilnear activation function that transforms the input in a point-wise fashion. This function is required to be a differentiable function. The functions are typically a Rectified Linear Unit(ReLU) or a Sigmoid function. For the AE also similarly we can write the minimization function for reconstruction of the input data given by the equation (2.2)

$$\min_{U,V} \|X - \sigma(XU)V\|_F^2 \tag{2.2}$$

In the equation (2.2), $\sigma(XU)$ denotes the low dimensional representation. The matrix U is a linear encoding and that should minimize the reconstruction loss. Amongst the ways to regularize the parameters of U and V, one of the popular ways is to apply constraints. The average of the activations in the latent layer is one of the constraints. This is in a form of enforcing sparsity.

2.1.2.3 Convolutional Autoencoders

In CAEs the main idea is to let the filters learn themselves like in a regular CNN but use the output of those filters to reconstruct the input image. The CAEs comprise two stacks as the firsts being the convolution stack and the second one being the transposed convolution or in other words the decoding convolution

stack. CNNs aim to classify the provided input while learning the filters by identifying features that are required for classification. CNNs are generally termed as supervised learning methodologies. But on the other hand, the task of CAEs is to learn filers that enable successful reconstruction of the inputs, instead of classifying into classes.

Convolutional AutoEncoders (CAEs) filter definitions are mostly manually engineered in terms of their number. Providing sufficient number of filters means learning the same number of distinct features and such learned filters can be reused in other computer vision tasks as well in the means of transfer learning and parameter initialization for the purpose of faster convergence and model guidance.

CAEs are the best method of learning convolutional filters in unsupervised manner. After a successful learning rounds these learned features can be utilized for feature extraction in new data sets. The latent space or the bottleneck layer is a compact representation of the input data and this can be very useful in many tasks including data compression where areas of data storage optimization and communication bandwidth reduction lie.

CAEs are more capable of learning features in comparison with AEs. The reason behind is that in AEs the parameters are global in all means and there is no concept of local features. Hence, this introduces parameter redundancy and the resource utilization is increasing exponentially for a small introduction of additional layer. This also constricts the models being able to handle larger inputs. But CAEs solves this issue due to their inherent structure of spatial locality. Here, the parameters are not global and in initial layers only the local features are leaned and gradually as it goes deeper, the learned features apply to the global scale.

The latent or bottleneck vector representation of the k^{th} filter for a single channel input x would be as in (2.3)

$$h^k = \sigma\left(\mathbf{x} * W^k\right) + b^k \tag{2.3}$$

The reconstruction is obtained by the latent maps $H(h^k \text{ for } k \in H)$ for decoding convolutional filter $\tilde{W}$.

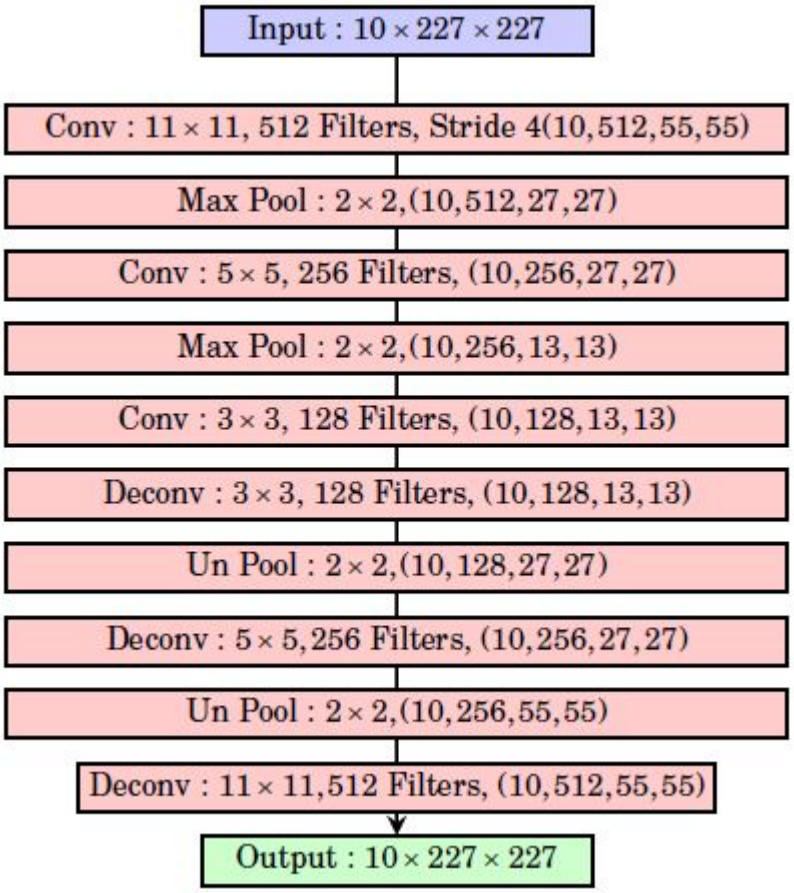
The σ function here is a non-linearity function that does point-wise operations. The bias values are broadcast into all the components of the latent map. Since there are several CAE layers are stacked together, the output of the first layer is input into the next layer.

In contrast with PCA, CAEs have some advantages. PCA ignores the spatial

structure and how the pixels are formed locally or in simple terms pixel location in the image. This is called permutation invariant. PCA introduces a large redundancy of parameters for considerably large images (100×100) and also it spans over the entire receptive field. CAEs have comparatively a lower number of parameters as the weights are being shared across many input locations.

In the recent research of anomaly detection, in [21] a deep CAE has been trained in a way when an input frames sequence is given, the network tries to reproduce the same sequence or fail with an error significant enough to separate out the anomaly. Spatio-Temporal Stacked frame Auto Encoder (STSAE) in [21] treats the image frames of each time slice and stack them together to form a sequence of p stacks. $x_i = [X_i\ X_{i-1}.....X_{1-p+1}]$ As mentioned, each slice is essentially treated as a different channel. The model is depicted in the figure 2.1.

Figure 2.1: Spatio-Temporal Stacked frame Auto-Encoder



The L2 regularized loss function is minimized over frames from the training

video.

$$L(W) = \frac{1}{2N} \sum_{i} \|\mathbf{x}_i - f_W(\mathbf{x}_i)\|_2^2 + \nu \|W\|_2^2 \tag{2.4}$$

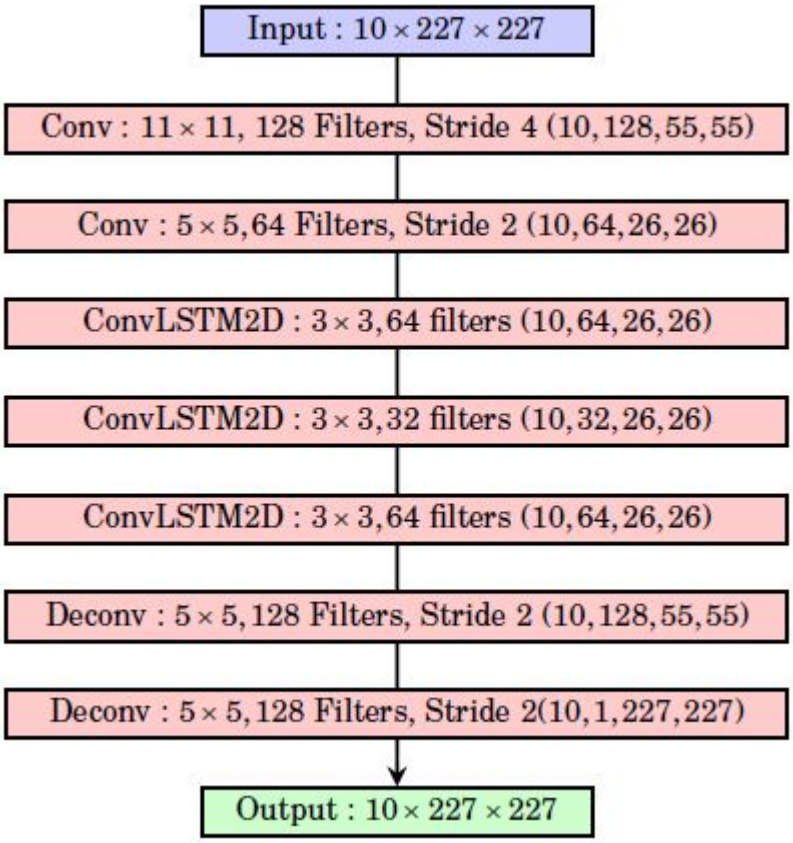
In the above equation, the tensor $x_i \in R^{r \times c \times p}$ is a 3 dimensional tensor with it's spatial dimensions being r, c and p being number of frames in the sequence upto p time steps back in time dimension, with hyperparameter ν that is used to balance the reproduction error and the norm of parameters. N is the size of the mini-batch.

In the model discussed in [19] is shown in the figure 2.2. The reconstruction error at the t^{th} time step is given by $E_t = \left| X_t - \hat{X}_t \right|$ while the regularity in terms of a score is given by the equation (2.5):

$$s(t) = 1 - \frac{\sum_{(x,y)} E_t - \min_{(x,y)} (E_t)}{\max_{(x,y)} (E_t)} \tag{2.5}$$

In this equation, the σ operators for $min_{(x,y)}$ and $max_{(x,y)}$ are directly upon the x and y spatial indices. One could either directly use the reconstruction error as the anomaly score or use Mahalanobis distance. This is evaluated as the error between the input and the reconstructed output from the AE.

Figure 2.2: Convolutional LSTM based autoencoder



2.1.2.4 Sparse Autoencoders

Sparse autoencoders are acting in a similar way as the normal autoencoder, but it has higher number of hidden units than the number of input neurons. Using a structure like that we can still find some interesting patterns in image sequences and can be used for anomlay detection purposes. This generally involves a sparsity penalty $\Omega(h)$ on the latent layer in addition to the reconstruction error.

$$L(x, g(f(x))) + \Omega(h) \tag{2.6}$$

Here $g(h)$ is the decoder input and $h = f(x)$ which is the decoder output.

In the research published by Medhini et. al. [23] describes a system that uses local and global descriptors that can identify an localize anomalies in a video frame. The local descriptors aid in learning local and temporal relationships

which is utilized for localizing anomalies and global descriptos constructed using deep sparse autoencoders are used for interpreting the video as a whole.

2.1.2.5 Contractive Autoencoders

The contractive autoencoders add the jacobian of the latent space representation to the reconstruction loss and and due to that, the latent space representation does not vary for comparatively smaller changes in the inputs [24]. If the latent space representation z is $z = f(x)$, and the decoder that maps to the input space, $r(x) = g(z)$ the regularized loss function can be written as,

$$L(W) = E_{\mathbf{X} \sim X_{\text{train}}} \left[L(\mathbf{x}, r(\mathbf{x})) + \lambda \left\| \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right\| \right] \tag{2.7}$$

Due to the regularized error function, the autoencoder becomes less sensitive to the input variation but enforcement of the minimal reconstruction error maintains sensitivity to the manifold having higher density. These type of autoencoders are sensitive to the variations along the manifold but not orthogonal to it, or in other words, it approximates the tangent plane of the data manifold. [25].

2.1.2.6 De-Noising autoencoders

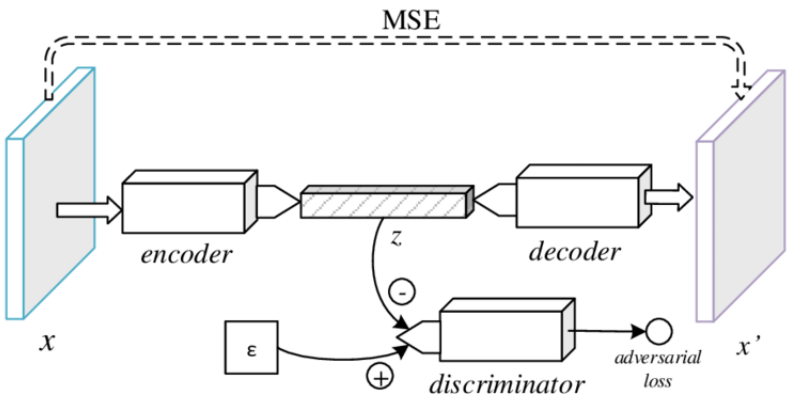
In the area of unsupervised learning, these autoencoders are vastly known for being one of the most robust feature extraction methodologies[26]. Instead of minimization of reconstruction error, reconstruction from the corrupted inputs is used. Stacked De-Noising Autoencoders (SDAE) are used to learn features from a frame sequence using both in terms of appearance and motion information. Appearance is captured using raw values and motion information is captured using the optical flow between consecutive frames[27]. These two types of SDAE pipelines are coupled together to learn a joint representation that captures both the above aspects.

2.1.2.7 Adversarial Autoencoder

Adversarial encoders are somewhat similar to variational autoencoders but in contrast to variational autoencoders the training process consists of an adversarial part that makes sure the latent space is regularized according to a given distribution. One advantage this has over variational autoencoder is that this has the ability to force any type of distribution over the latent space where in

variational autoencoder the latent space had no option but to follow a normal distribution.

Figure 2.3: Adversarial Autoencoder



As an autoencoder the encoder and decoder combination tries to reconstruct the original input image as closely as possible using an error function, usually mean squared errorMSE or mean absolute errorMAE.

The equation for the autoencoder part is as below. This is exactly same as a usual autoencoder functionality.

$$\mathcal{L}_{rec(E,G)} = \frac{1}{2N}\sum_{i}^{N}(\mathrm{x}_i - \mathrm{x}'_i)^2 \tag{2.8}$$

The goal of the AAE is to learn the two functions (E,D) where $x' = E(G(x))$ is close to the original input x

The latent vector regularization is performed in a adversarial fashion where the optimization function is as below.

$$V(E,D) = \min_{E}\max_{D} E_{\mathbf{z}\sim p_z}[\log D(\mathbf{z})] + E_{\mathbf{x}\sim p(\mathbf{x})}[\log 1 - D(E(\mathbf{x}))] \tag{2.9}$$

The function $V(E,D)$ is solved while aiming to distinguish the distribution $q(z \sim E(x))$ from prior distribution $p(z)$

2.1.2.8 Deep Belief Networks

These type of networks are created by stacking multiple hidden layers. Restricted Boltzmann Machines RBMs are stacked and trained in a greedy fashion to con-

struct Deep Belief Networks (DBN). Those are trained in an unsupervised manner in a greedy fashion to perform feature learning. They are capable of reconstructing the original inputs and thus called generative models.

In [28] the DBNs are used to represent raw image representations. The researchers have proposed a unified energy-based methodology for video abnormality detection. Their model is based on RBMs (DBNs) to capture data irregularity. Their system can distinguish and pinpoint the anomaly in the spatial plane. It is trained directly on the image in a fully unsupervised manner. For video streaming, they further introduce a streaming type methodology that can update parameters in the real time incrementally while a video frames are being input.

2.1.3 Predictive Modeling

If the current output frame at time t is X_t, the basic idea of predictive modeling is that to represent the current frame in terms of past p frames $[X_{t-1}, X_{t-1}, ..., X_{t-p-1}, X_{t-p}]$. This principal is used in auto-regressive models for time series analysis which employs a linear function over the past data. The same is used with non-linear functions such as sigmoid functions in Recurrent Neural Networks(RNN), modeled as recurrent relationships. The standard method for these type of modeling is the LSTM which in fact is a extended RNN that has a gating functionality introduced in order to cater for the issue of vanishing gradient that was present in normal RNNs when they were subject to backpropagation through time. Recently there have been some researches done on the video prediction domain using convolution networks by minimizing mean squared error(MSE) between predicted and future frame [29]. A similar research is [30], which uses a CNN-LSTM-deConv network by combining mean sqared error and adverserial loss.

2.1.3.1 Usages of LSTM (Long-Short-Term-Memory)

LSTMs are capable of remembering past frames while resolving issues aroused by vanishing gradient. This is performed by enabling various gate mechanisms that allow it to filter and carry forward the required information and retain only the necessary part. LSTM principal is used combined with other models as well. One such successful combination is Convolutional LSTM, which is a LSTM network formed in a 2D convolution network pattern.

2.1.3.2 Convolutional Long Short Term Memory (ConvLSTM)

This is an encoder model that is a composite of the LSTM model and the encoder decoder model. The fully connected LSTM is a powerful model but it is too redundant for spatial data. Convolutional LSTM takes a encoding -forecasting structure that has stacks of Concolutional LSTM layers.

This model is basically designed to overcome the major drawback of fully connected LSTM models where they are densely connected in terms of input-to-state and state-to-state transitions. ConvLSTM is nothing but adding a convolution operator between state-to-state and input-to-state transitions.
The operations of a ConvLSTM are represented by the equations shown below. The convolution operation is denoted by $*$ and the Hadamard product is denoted by $\circ$ (the element-wise product of matrices).

$$\begin{aligned} i_t &= \sigma\left(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i\right) \\ f_t &= \sigma\left(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f\right) \\ \mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh\left(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c\right) \\ o_t &= \sigma\left(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_t + b_o\right) \\ \mathcal{H}_t &= o_t \circ \tanh\left(\mathcal{C}_t\right) \end{aligned} \tag{2.10}$$

The encoding network compresses the input tensor to the hidden state and the forecast network unfolds the hidden state network to make predictions. Hidden representations can be utilized to capture mobile objects as used in the this research, where the main objective would be to capture moving pedestrians. If a larger kernel is used, faster moving objects can be captured and if a smaller nucleus is used, slower objects can be captured [31].

An encoder decoder model is used in the researches performed in [32] and [33] with some outstanding results. In their models, a ConvLSTM model was used along with an encoder decoder model as a unit in a composite LSTM model, with two branches, where one reconstructs the input and the other tries to predict the future inputs.

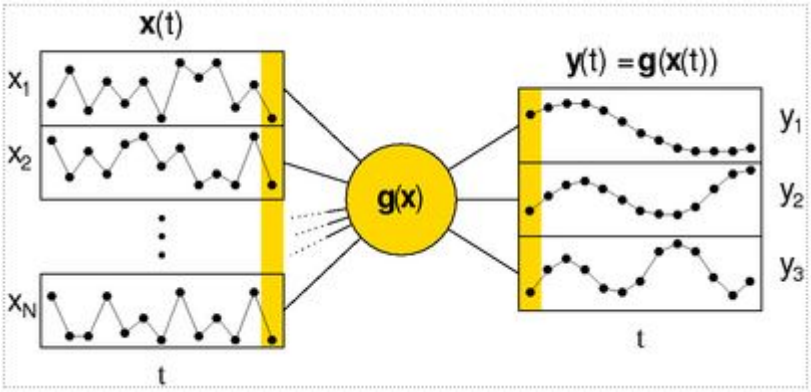
2.1.3.3 Slow Feature Analysis (SFA)

Slow feature analysis is based on the principal of slow features. In a series of image frames that change over time, or in other words, in a video frame, there are slowly moving features that can be captured. Even though the individual pixel values change drastically, and more frequently over time, these slow features are

not changing as frequent as much as in each pixels. These higher order internal visual representations very slow on time scale and identifying these would be useful in prediction of the future inputs and thus formulating a anomaly detection framework. This is entirely based on the slowness principal.

The Slow Feature Analysis algorithm formulates the general intuition behind the slowness principle in terms of a non-linear optimization problem : Given a input signal $x(t)$, find $g_j(x)$ functions such that the output signals. This is an optimization problem which is denoted by the following equations.

Figure 2.4: SFA



$$y_j(t) := g_j(\mathbf{x}(t)) \tag{2.11}$$

Optimization problem

- Minimize : $\Delta(y_j) := \langle \dot{y}_j^2 \rangle_t$

Constraints:

- Zero mean : $\langle y_j \rangle_t = 0$
- Unit variance : $\langle y_j^2 \rangle_t = 1$
- De-correlation of different signal outputs : $\forall i < j : \langle y_i y_j \rangle_t = 0$

The constraints enforce the representation to have a unique solution and unit covariance to avoid trivial zero solution. Also uses de-correlation of the feature to avoid redundancy in them.

In [34] and [35] SFA has been used for pattern recognition. An incremental updating mechanism of slow features was introduced by the authors in [36]. The

SFA here is calculated using the batch PCA method by iterating two times. In order to produce a traceable solution, another two layered localized SFA architecture is introduced by the researches who performed [37].

2.1.3.4 Other Predictive Models

2D convolution nets are appropriate for image recognition tasks, but they are not a good fit in terms of detecting the information in the temporal dimension encoded in subsequent frames in video sequence analysis tasks. As a solution, the authors in [38] are using a 3D convolutional architecture, which is used in the layers of an autoencoder. Such an autoencoder is capable of learning features that are invariant to spatial and temporal changes or in other words, mobility, encoded by 3D convolutional feature mappings. The kernel is a 3d tensor and the output of such a kernel is also another 3D tensor with one temporal dimension which is expected to encode motion related information. Authors in [39] have proposed a 3D kernel that was created by stacking T-frames on top of each other, as in [21].

Another prediction model is used in [40] in which a CNN's features were input to a LSTM model so as to predict the formed latent layer representation. The obtainable prediction error was employed to evaluate novelties in applications related to robotics. In [41] authors have attempted to create a video forgery detection system by using a recurrent autoencoder that utilizes an LSTM which is used to model the temporal dependency between patches from a sequence of video frames.

2.1.4 Deep Generative Models

If a supervised learning setup $(X_i, y_i) \in R^d \times \{C_j\}_{j=1}^K$ is indexed by i where $i = 1 : N$ in the data set, a generative model estimates the class conditional posterior distribution $P(X|y)$. Training this could become unstable if the input data is high quality images or spatio-temporal tensors.

2.1.4.1 Variational Autoencoders

Variational Autoencoders basically models the data distribution $P(X)$ of a high dimensional input X which could be an image or video. A encoder decoder architecture is used with some parameters θ and ϕ which essentially achieves the variational approximation of the latent space.

The goal of a variational autoencoder is to learn a dimensionaly reduced representation z by modeling P(X—z) with more simpler distribution, usually a Gaussian, i.e. $P(X|z;\theta) = N(X|f(z;\theta), \sigma^2 * I)$. The loss function has two terms. One is the reconstruction error and the other is the KL-divergence term. Anomaly detection using variational autoencoders is experimented in [42]. They evaluate the reconstruction probability $E_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]$. First, for a new sample $x^{(i)}$ the mean and the standard deviation vectors are evaluated with the encoder, $(\mu_{z(i)}, \sigma_{z(i)}) = f_\theta(z|x^{(i)})$. Then the latent space vectors are sampled, $\boldsymbol{z}^{(i,l)} \sim N\left(\boldsymbol{\mu}_{\boldsymbol{z}^{(i)}}, \boldsymbol{\sigma}_{\boldsymbol{z}^{(i)}}\right)$. Then the parameters for the input distribution are reconstructed using the L samples, $\boldsymbol{\mu}_{\hat{x}^{(i,l)}}, \boldsymbol{\sigma}_{\hat{\mathbf{x}}^{(i,l)}} = g_\phi(\mathbf{x}|\mathbf{z}^{(i,l)})$. Thus the reconstruction probability for the sample $x^(i)$ is given as follows.

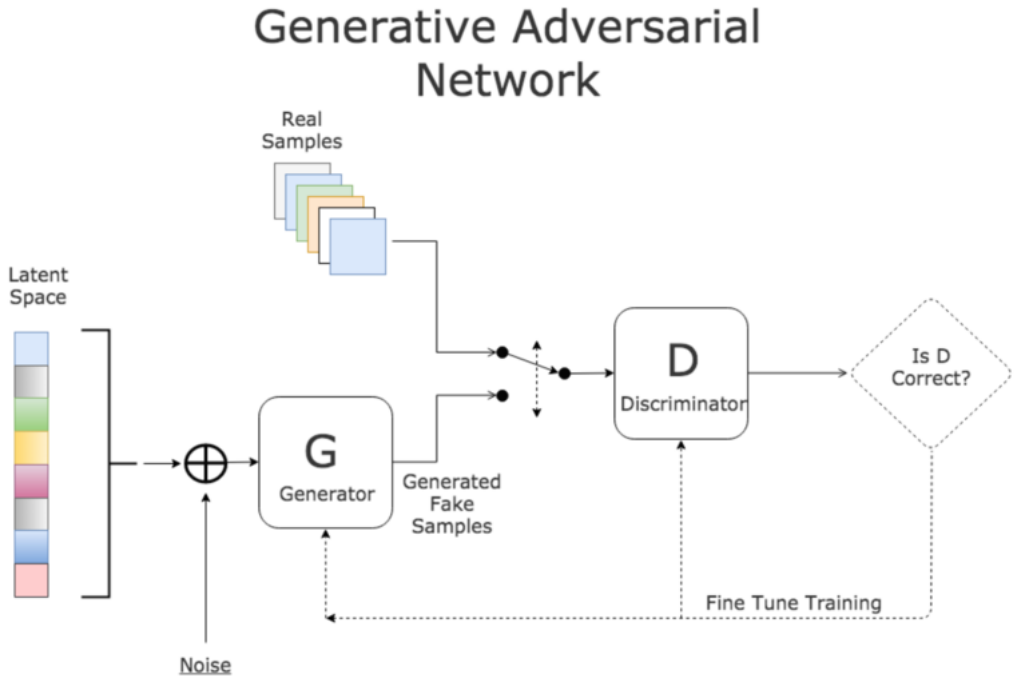
$$P_{\text{recon}}\left(\mathbf{x}^{(i)}\right) = \frac{1}{L}\sum_{l=1}^{L} p_\theta\left(\mathbf{x}^{(i)}|\boldsymbol{\mu}_{\hat{\mathbf{x}}^{(i,l)}}, \boldsymbol{\sigma}_{\hat{\mathbf{x}}^{(i,l)}}\right) \tag{2.12}$$

In a variational autoencoder one of the main distinctions from a standard autoencoder is that, sampling the latent variable distribution lets the variability to be taken into account by the reconstruction error rather than in a case where latent variables are defined by deterministic mappings.

2.1.4.2 Generative Adversarial Network(GAN)

GANs were initially introduced by researches in [43]. The GAN contains a generator G which is a decoder usually and a discriminator which is usually a encoder. The task of the generator is to learn a distribution p_g over the data x by mapping $G(Z)$ of the samples z to 2-dimensional images in the image space manifold X. The image space X is populated by regular data samples. Here, z is a 1D vector from the input noise that is uniformly distributed and sampled from the latent space Z. The generator G is generally a convolutional decoder but there are some variations to this where this may not necessarily be a correct term. In researches like [44](Adversarial Dual Autoencoder(ADAE)) the generator is an autoencoder which takes in an input image instead of the latent vector z.

Figure 2.5: The basic structure of a GAN



The discriminator is usually a CNN that maps the 2D input to a signal score. The discriminator output is an interpreted as the probability (a value between 0 and 1) which indicates whether the given input is a real image sampled from X or a generated image faked using G(z) by the generator G. Again there can be variations to the discriminator as well. The best example is that the [44] in which the discriminator is also an autoencoder where the reconstruction error us used instead of the likelihood score which was automatically generated by an ordinary discriminator.

Discriminator D and Generator G are both optimized simultaneously through the minimax function mentioned below.

$$\min_G \max_D V(D,G) = E_{\mathbf{X}\sim p_{\text{ data}}}[\log D(\mathbf{x})] + E_{\mathbf{z}\sim p_z(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))] \quad (2.13)$$

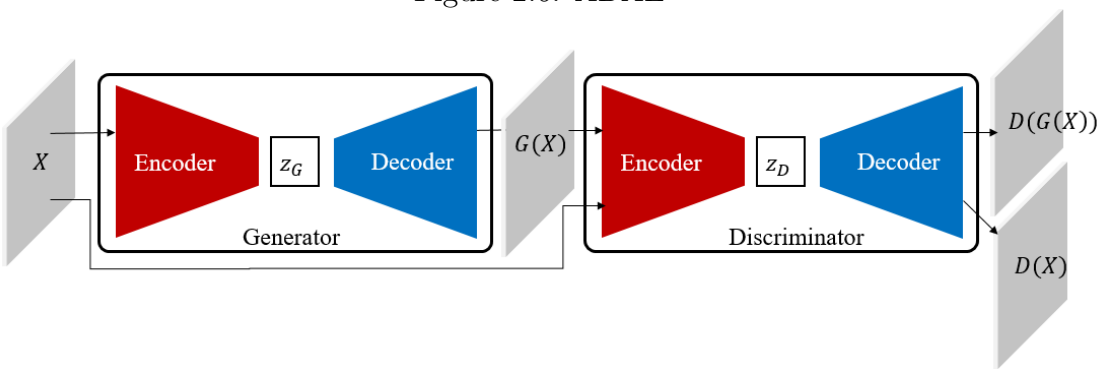
The discriminator is optimized to give the maximum probability for "real" for those samples that are sampled from the real samples and "fake" to the samples that are generated by the generator. The main objective of the generator G is to minimize $\log(1 - D(G(\mathbf{z})))$ by maximizing $D(G(\mathbf{z}))$, thus it essentially tries to fool the discriminator D by doing so.

In [45] a GAN model was applied for the task of identifying the anomalies in medical images. GANs are generative models that are capable of generating training data points $\mathbf{x} \sim P_{\text{data}}(\mathbf{x})$ where P_{data} represents the probability density of the training data points.

For anomaly detection, the GANs can be used in many different ways. One is to threshold the final output of the discriminator for each input for the generator. The other method is that to consider a score like MSE for the output of the well trained generator output. Another interesting way would be to separate out the discriminator and use it as a judge for other different types of reconstruction methods like CAEs. If deformity in the reconstructed image are visible and detectable, the separated discriminator can be used out take on the reconstructed output and provide judgement.

In order to obtain a likelihood in GANs, a mapping from the input image domain to the latent domain is required. The authors in [45] have made a success in creating this mapping. In this research, the authors attempt to find a point z in the latent vector space that corresponds to an image $G(z)$, which is close to the image x and this is located in manifold X. The similarity of x and $G(z)$ depends on how much the input image adheres to the data distribution p_g, on which the generator was trained upon.

Figure 2.6: ADAE



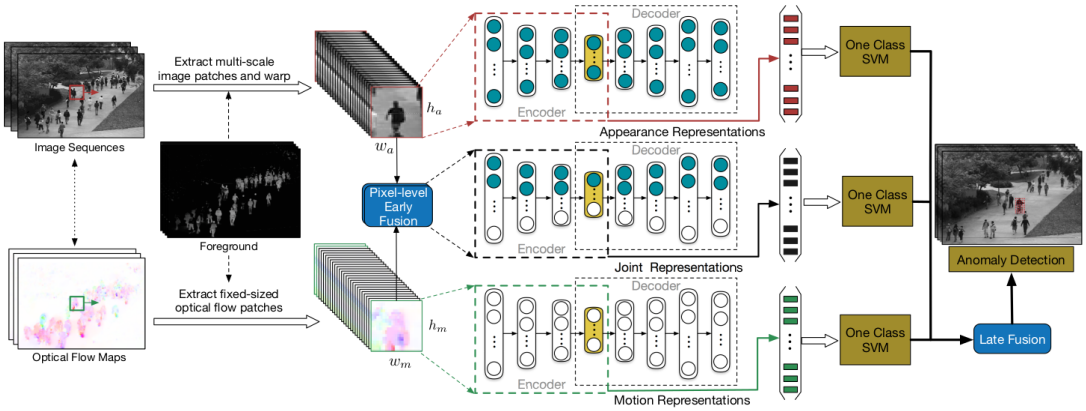
2.2 State-Of-The-Art Models

2.2.1 AMDN

This is the model proposed by [46] in 2015. Currently this model performs well for both PED1 and PED2 but specifically provide state-of-the-art AUC/EER for PED1.

This method uses an optical flow base score and raw image score combining method called double fusion.

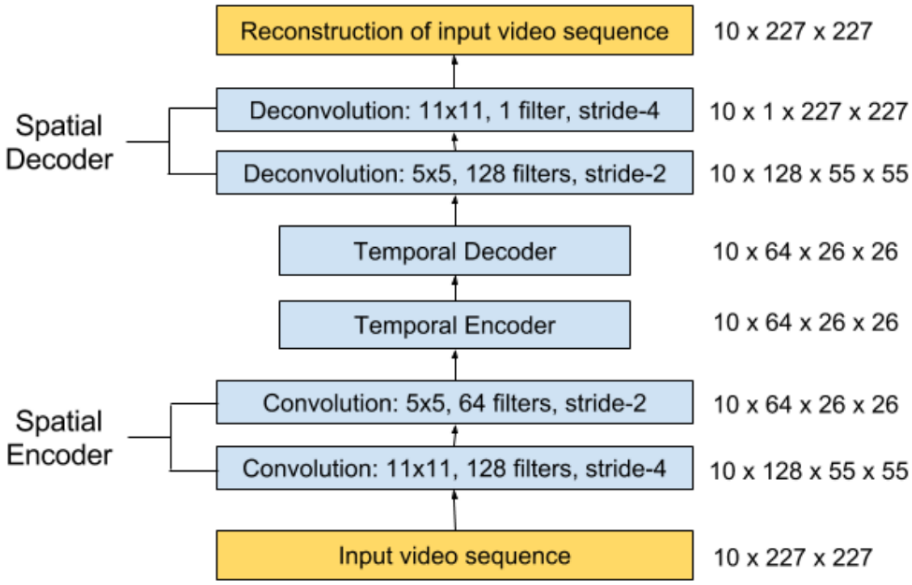
Figure 2.7: AMDN



2.2.2 ConvLSTMAE

This model was introduced in 2017 by [19]. Provides it's best performance on PED2 and currently holds the state-of-the-art AUC/EER for PED2. Some LSTM layers are used.

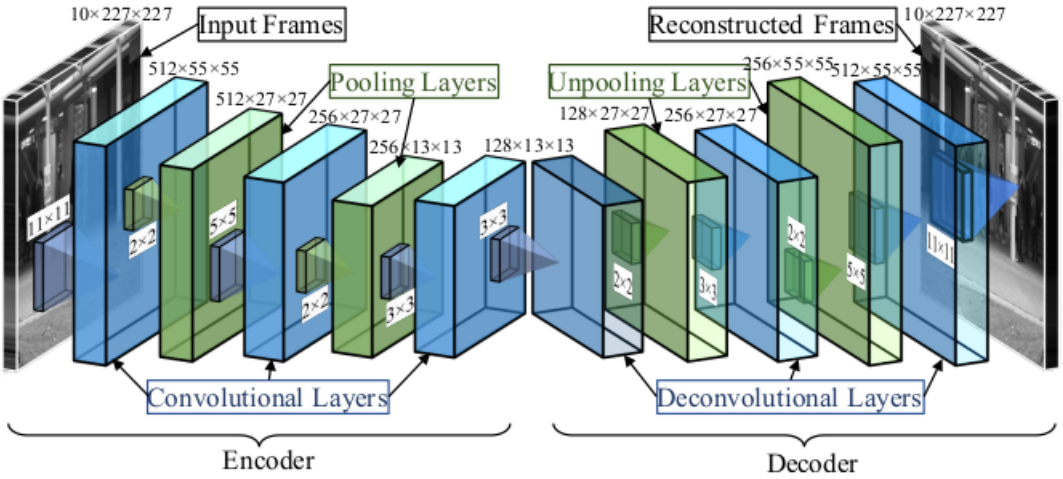
Figure 2.8: ConvLSTMAE



2.2.3 ConvAE

This model is an autoencoder which was introduced by [21] in 2016. This model does not have LSTM layers for temporal data capturing. Used HOG+HOF features.

Figure 2.9: ConvAE



Chapter 3

Methodology

3.0.1 Model Architecture

This section describes the architecture and metrics that were used to train the model. The current literature in the domain directed the search towards deep learning as deep learning has become the new star in the area of video processing. The non-deep learning methods were very useful till the time that processing power was still an issue. But since processing power is available at a lower cost than before, it is always worth putting efforts in deep learning. On the other hand, deep learning is giving promising results that were not achievable through non-deep learning methods. In terms of accuracy and robustness deep learning proves to be the best. The only drawback is that it needs a huge amount of data to gain such better accuracy. Data is at abundance and with the huge amount of data at hand, a large processing power is also a requirement. Even though processing power is cheaper than before, it is still a considerable cost. If the target architecture is a deep learning based method, it should also be feasible in terms of cost of processing.

3.0.1.1 Motivation

This system is a unsupervised system which is hard to achieve using the conventional non-deep learning based methods. Hence the only choice left is to use a deep learning method that can be trained without any supervision.

The most common way of anomaly detection in the deep learning world is the use of autoencoders. Since the introduction of the methodology in [19], there have been many variations to the methodology. Even though [19] has demonstrated the methodology using the UCSD crowd data set, most of the later introduced methodologies were introduced using the MNIST data set. MNIST is a very basic data set of which the images are of 28x28 dimensions and no continuity is expected but the mere image pattern observation. But in contrast, UCSD data set is more complex (dimensions 158x238). Not only it has a higher resolution, but also it has its anomalies in sequences which requires adding a temporal dimension into our equation. Due to these two reasons, the the input data becomes 3 dimensional and requires a higher number of parameters in the model to be trained. This very thing makes it extremely difficult to train on GANs which are already very unstable in nature. Experimentation with GAN architectures showed that the models are highly unstable with such larger volumes and always tend to drop into a local minima. The only architecture that shows a appreciable level of robustness towards such high dimensional data is the autoencoder and

any variation of autoencoders seemed to be able to train and obtain good results.

Among the autoencoder types, two of the most promising types are adversarial autoencoder and the variational autoencoder. Even though both try to approximate the latent space using a distribution, adversarial autoencoder facilitates a few additional flexibility points that are to be noted. The main point is that we can use any distribution to be followed by the latent space where as in variational autoencoder it is only a normal distribution. This quality can be utilized and different distributions would show different results on the learned error rate. The other advantage is that the AAE has a part that is trained in adversarial fashion. An adversarial discriminator is nothing but an improved error function that cannot be designed by human intelligence. Hence, AAE provides the chance of utilizing an improved error function, which in turn can also be used as an anomaly score. This anomaly score would be able to detect even small anomalous behaviours in sequences where a general error function like MSE or MAE would not be very sensitive to them. But this outputs very fluctuating non-smooth curves which may need a lot of effort spent on curve smoothing and threshold decisions. Attributing to the sensitivity to even small variations of the input, its quite common to see many false positives if the anomaly score is taken solely based on latent dimension. Hence the adversarial training on the latent dimension is only performed as a regularization methodology. Discriminator's output was not directly taken as the anomaly score. Instead the reconstruction error from the main decoder was taken as the anomaly score. Due to the additional regularization step, this score would yield smoother curves, that are smoother than in [19]. In addition to that, the research carried out by [47] claims that the reconstruction outputs obtained by AAE are sharper than what was obtained by VAE. VAE reconstruction outputs are sharper in quality than from an AE due to the fact that latent space being continous. Since the research by [47] indicates that AAE gives better quality than VAE, AAE automatically becomes a sharper results producer than a regular AE.

In [22], the researchers have used an AAE to detect anomalies and the methodology that they had used lacks a few points. The anomalies are not always visible in a single image alone and observing patterns in a single image would not suffice to detect anomalies of a more complex context. An example would be two people fighting on the pathway. In order to detect these type of anomalies, there is a need to capture the anomalous behaviour in the temporal dimension, which essentially suggests the use of a recurrent network, an LSTM to be precise. Since there is also a spatial dimension to be considered, the most suitable would be to

use a Convolutional LSTM network.

For this research the model used is an adversarial autoencoder. Instead of using convolutional layers alone, there are some colvolutional LSTM layers added so that the history is efficiently managed to reproduce in a predictive manner. The expectation is that whenever the input contains abnormalities, the reconstructed image would be reproduced in a deformed manner. This deformity is also detectable form observing the latent space.

3.0.1.2 Model Definition

Layer	Filters	Kernel	Stride	Normalization
TimeDistributed(Conv2D)	128	11x11	4	Layer Normalization/ReLU
TimeDistributed(Conv2D)	64	5x5	2	Layer Normalization/ReLU
ConvLSTM2D	64	3x3	1	Layer Normalization
ConvLSTM2D	32	3x3	1	Layer Normalization
Flatten - Latent Vector	327680	-	-	-

Table 3.1: Encoder Network

The encoder network consists of time distributed convolution layers and convolutional LSTM layers.The original image sequence of dimensions (10, 256, 256, 1) is given as input. The latent vector of size 327680 is given as the output.

Layer	Filters	Kernel	Stride	Normalization/Activation
Flattened Latent Vector	327680	-	-	Reshaped (10, 32, 32, 32)
ConvLSTM2D	32	3x3	1	Layer Normalization
ConvLSTM2D	64	3x3	1	Layer Normalization
TimeDistributed(Conv2D)	64	5x5	2	Layer Normalization/ReLU
TimeDistributed(Conv2D)	128	11x11	4	Layer Normalization/ReLU
TimeDistributed(Conv2D)	1	11x11	-	sigmoid

Table 3.2: Decoder Network

The decoder network consists of time distributed convolution layers and convolutional LSTM layers. The latent vector of size 327680 is given as the input. Original input dimensions (10, 256, 256, 1) as output

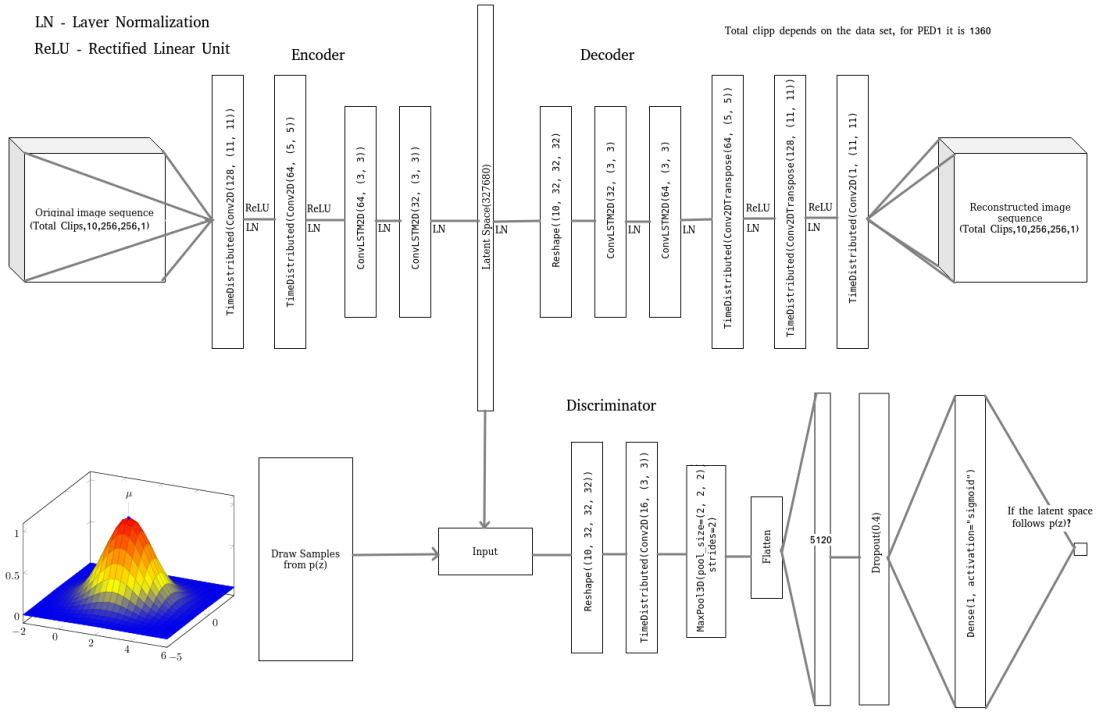
Layer	Filters	Kernel	Stride	Normalization/Activation
Flattened Latent Vector	327680	-	-	Reshaped (10, 32, 32, 32)
TimeDistributed(Conv2D)	16	3x3	2	Relu
MaxPool3D	128	2x2x2	2	-
Flatten	24576	-	-	-
Dropout(0.4)	-	-	-	-
Dense	1	-	-	sigmoid

Table 3.3: Discriminator Network

The discriminator network consists of 2D convolution layers Dense Layers. The latent vector of size 327680 is given as the input. Dense layer with sigmoid activation as output

The model layer descriptions are as depicted in tables 3.1 , 3.2 and 3.3.

Figure 3.1: Overall Model Architecture



3.0.1.3 Training

The training setup consists of two parts. First some level of pre-processing had to be performed before applying the training algorithm. The images were first resized to 256x256 size from their original sizes and normalized to have values between 0 and 1. in [19] the pre-processing step was to resize them to 224x224 resolution. In the proposed model, the resolution was slightly increased expecting better detection of the deformities at reconstruction.

On Ped1 and Ped2, the model was trained for 100 epochs each and with batch size of 4. The input is of dimensions, (10, 256, 256, 1). Images were divided into sequences of 10 and created clips of the above shape. Since the data set was not sufficiently large for a deep learning model, as a data augmentation technique, clips were created using variable strides of 1, 2, and 3.

The encoder and decoder were trained without any adversarial effect. The loss were the reconstruction error that was measured by MSE. Then the encoder vs discriminator training was performed in an adversarial manner. The error used was binary cross entropy. When the discriminator is trained, a vector that matches the same size as the latent vector is given as the input to the discriminator. This vector is sampled form a prior $p(z)$ such that $z \sim \mathrm{N}(0, 1)$.
The optimizer used was the Adaptive Moment Estimation(Adam) with learning rate 10^{-4}, decay 10^{-5}, and epsilon 10^{-6}.

Reconstruction error between the generator and decoder was Euclidean distance between input and reconstructed images.

$$e(t) = \|x(t) - f_W(x(t))\|_2 \tag{3.1}$$

f_W represents the model weights of encoder-decoder network. The reconstruction error of all pixel values in frame t of the video clip is calculated using the Euclidean distance as in (3.1).

3.0.1.4 Anomaly Detection

The abnormality score is calculated based on the (3.1).

$$s_a(t) = \frac{e(t) - e(t)_{\min}}{e(t)_{\max}} \tag{3.2}$$

This error function is as same as used in [19].

And then the anomaly score $s_a(t)$ is converted to a regularity score $s_r(t)$ so that the more anomalous the data, the lower the score would become.

$$s_r(t) = 1 - s_a(t) \tag{3.3}$$

In order to make the graph smooth so that threshold intersection points are easily measured, Persistance1D(.3) algorithm was used.

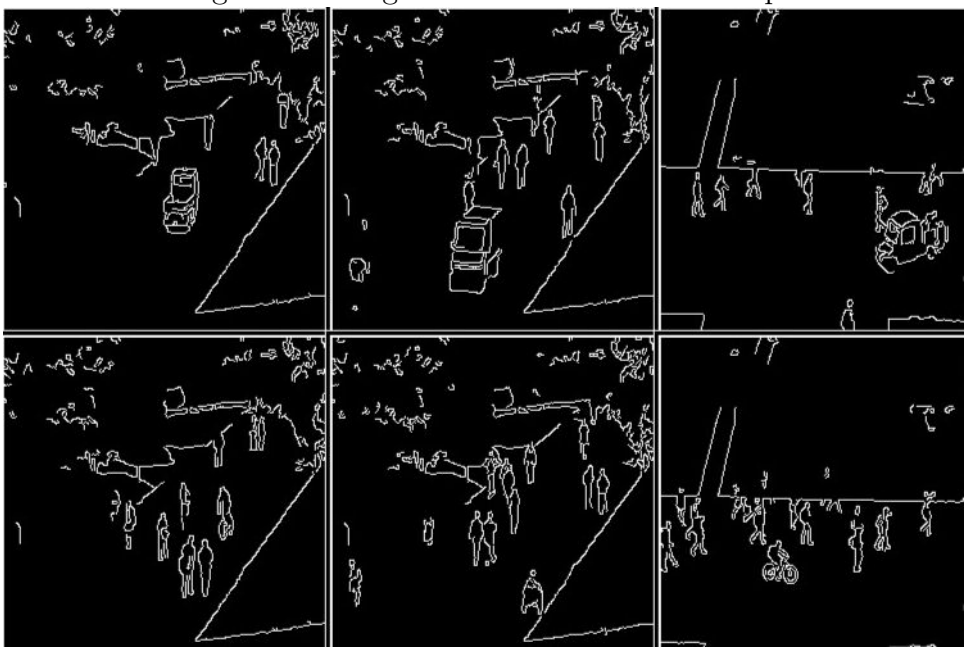
3.0.1.5 Variations of Anomaly Score Calculation

The model is three-fold in terms of anomaly score calculation. The method varies with the mode of weighting the individual pixels. The weights for the pixels are applied identically to both input reference and reconstructed frames. The three types are, the vanilla model, the model with edge based weighting, the model with background subtraction based scoring. Each of the three forms perform differently on the PED1 and PED2 data sets and this is explained in greater detail under the results section.

In the vanilla model, no weighting was used for reconstruction error calculation. It is the pure output of the model that is trained.

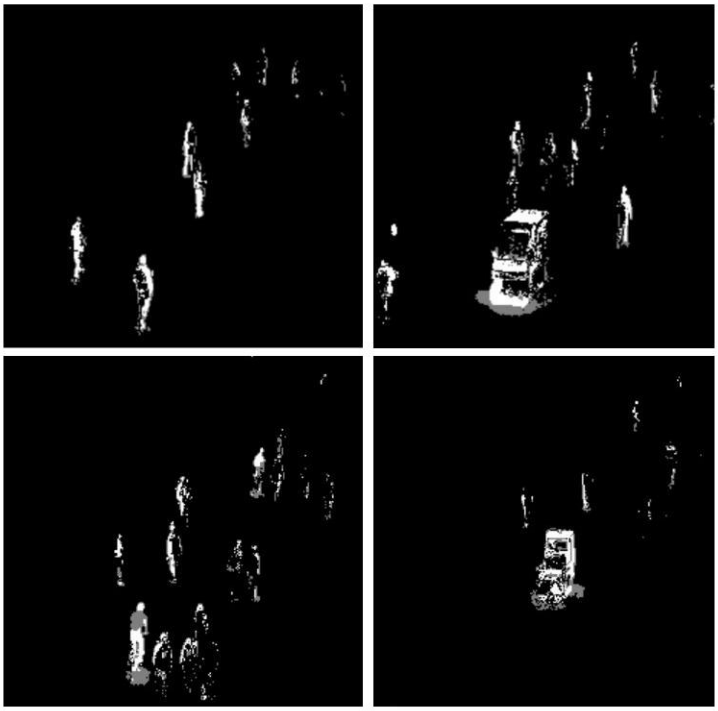
In the edge based weighting methodology, the weights are increased along the edges and this makes the model react to only object margins and internal textures of the objects are not evaluated. This way, the deformations of the reconstructions are better captured.
The used edge detector is 'Canny' edge detection'(.1)

Figure 3.2: Edges found in PED1 video clips



In the background subtraction based system, the anomaly score is improved by calculating the score only on moving areas. MOG2 background subtraction method is used to perform the background removal process. For the detected foreground area, a higher weight is given so that the reconstruction error stands more sensitive to the deformations and it becomes more robust towards the unnecessary noise.
The used background subtraction method is MOG2(.2)

Figure 3.3: Background Subtraction on PED1 video clips



Unlike in 3.2, in 3.3 only the moving objects are visible. This enables the system to manage the reconstruction error more effectively.

Chapter 4

Experiments and Results

4.1 Data Set

The research is performed on two of the most cited and widely used data sets, namely UCSD crowd anomaly data set, PED1 and PED2 where PED1 is about twice the size of PED2. PED1 has 34 training clips and 36 testing clips, each video containing 200 frames. In PED2 16 training and 12 testing videos are present and each clip contains a variable number of frames. The videos are created with a fixed position camera for each data set. PED1 has been filmed over-watching the crowd walking towards and away from the camera at a distance while PED2 has been filmed watching the crowd walking parallel to it. The training data only consist of normal clips and no abnormal event is included. Testing videos on the other hand has abnormal behavior varying over a number of classes. The categories as bikers, skaters, carts, wheelchairs and people walking in the grass area as well as anomalous pedestrian motion patterns.
In order to enhance the data set to make it more suitable for data hungry deep learning tasks, each video is divided into frames of 10 sequences and the 10 frame sequences were gathered over the same video using a stride varying from 1 to 3. This was able to increase the data set size by 3 times.

4.2 Experiments

The research is performed focusing on two models. The Ped1 and Ped2 of UCSD crowd data set are commonly used by the vast majority who are experimenting on crowd anomaly detection. Hence it is easy to compare performance with other models. All three types of models(Vanilla, edge weighted, background subtraction based weighting) are extensively tested on the PED1 and PED2 data sets.

This research has also performed experiments on the CHUK Avenue data set, which is another data set that is filmed using a fixed camera viewing the activities near a campus entrance. This data set is not cited by all the researchers but only a few. Thus Avenue data set was trained and tested as an extended experimentation task. This data set does not have a variety of anomalous classes, instead this has classes mainly runners, walkers towards camera, and throwing objects.

Further experiments focus on the possibility of utilizing the latent space information and the discriminator decision to detect the anomalies instead of examining the reconstructed image.

4.2.1 Hardware Settings

The hardware that were used are as below. The models were run on google collab services using Keras with tensorflow-2.0 back-end with the following specifications.

- NVIDIA-SMI 440.64.00 : Driver Version: 418.67
- CUDA Version: 10.1
- Tesla P100-PCIE
- RAM : 16280MiB

4.2.2 Anomaly Count

Anomalous events are detected based on the following criteria.

- A predefined threshold is used to capture the anomalous events
- If there is a local minima that has a lower value than the threshold, the local maximas on either sides enclosing the particular local minima are considered as the endpoints of the range of detected frames.
- If the detected range covers at least 50% of the frame span of the ground truth, the detection is considered as a true positive
- The coverage is considered as the accumulated coverage area of all the local minimas that has a lower value than the threshold
- For a detection to be a true positive, only the accumulated coverage from local minimas residing within the ground truth range are considered. Or at least the local minima should lie within 10 frames of either sides of the ground truth bounds.
- Similarly, if the accumulated coverage of local minimas is covering at least 50% of the non-ground truth areas, the detection is a false positive.

4.2.3 Experiment Methodology

All the test cases are run through the adversarial autoencoder and scores are calculated as per (3.2), and (3.3). In [22], they have used a adversarial autoencoder, but they have utilized only the discriminator score. Discriminator score is a useful piece of information, but since it looks at the distribution of the latent dimension, this will not have the capability to distinguish a skateboarder from a walking passenger. But it would be very useful in terms of understanding anomalies in the static background. For instance a passenger entering the grass area outside the pathway would be successfully flagged. Also different types of vehicles would be giving a considerable score in the latent vector evaluation.

In this research, the score is evaluated by calculating (3.3) and evaluating against an appropriate threshold value.

Int [19], they use the decoder reconstruction based score. In [22], they use latent latent vector evaluation. In [22] the the score fluctuates largely and the robustness and the score consistency is affected by it. Hence, the proposed systems do not use the latent information for anomalous score calculation, instead the reconstruction score error is used. But for the purposes of experimentation, the same is tested in a few test cases.

The testing was done using the test clips provided in PED1 and PED2 data sets in which they have provided the default set of ground truths. But the clips contain other anomalies that were not included in the ground truths. Hence more experiments were performed using the corrected ground truths.

4.3 Results

This section discusses the results obtained by the proposed systems(all 3 types) and they are compared with the other best performing models. The results show that the proposed systems outperforms the other systems in frame level detection tasks in terms of AUC of the ROC curve and the EER.

Method	PED1		PED2	
	AUC	EER	AUC	EER
Adam[48]	77.1	38.0	-	42.0
SF[49]	67.5	31.5	55.6	42.0
MPPCA[50]	66.8	40.0	69.3	30.0
MPPCA+SF[50]	74.2	32.0	61.3	36.0
HOFME[51]	72.7	33.1	87.5	20.0
ConvAE[21]	81.0	27.9	90.0	21.7
ConvLSTMAE[19]	77.1	27.5	[93.0]	13.0
AMDN[46]	[92.1]	16.0	90.0	17.0
Proposed(Vanilla)	79.4	23.9	97.6	10.0
Proposed(Edge Weighting)	80.6	26.9	94.0	16.9
Proposed(BG Sub Weighting)	85	22.4	94.6	12.8
Proposed(BG Sub Weighting - Ground Truth Corrected)	91.6	13.28	-	-

Table 4.1: The table lists the AUC(Area Under Curve) and EER(Equal Error Rate) rates of state of the art models from 2010 to 2019. Along with listed are the corresponding rates of the proposed model. Higher the AUC the better and lower the EER the better. The current state-of-the-art rates are in square brackets.

In the illustrations below, 'data index' refers to frame number and 'data value' refers to normality score value.

4.3.1 UCSD PED1

Figure 4.1: ROC for PED1, vanilla model with uniform weighting

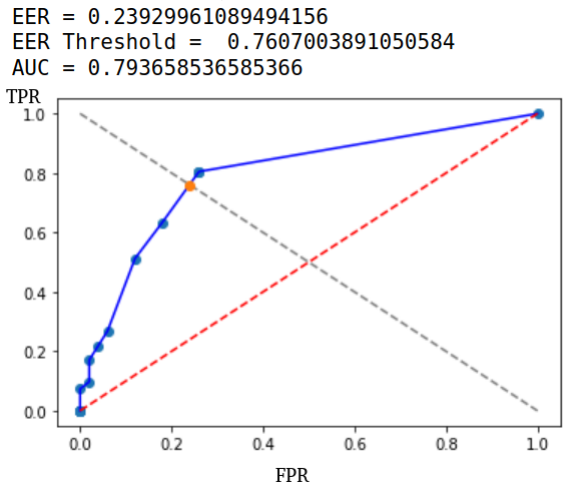


Figure 4.2: ROC for PED1, edge based weighting

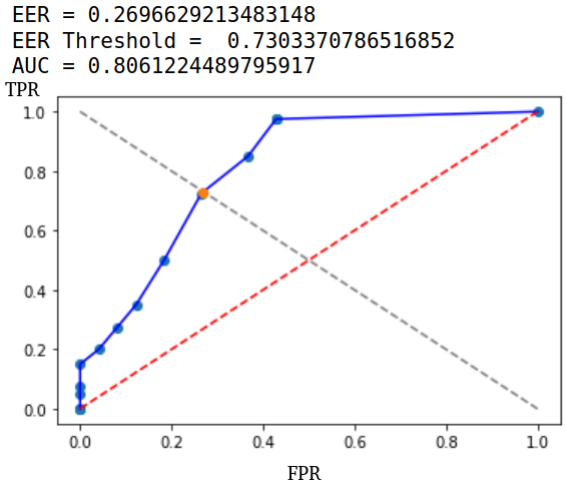
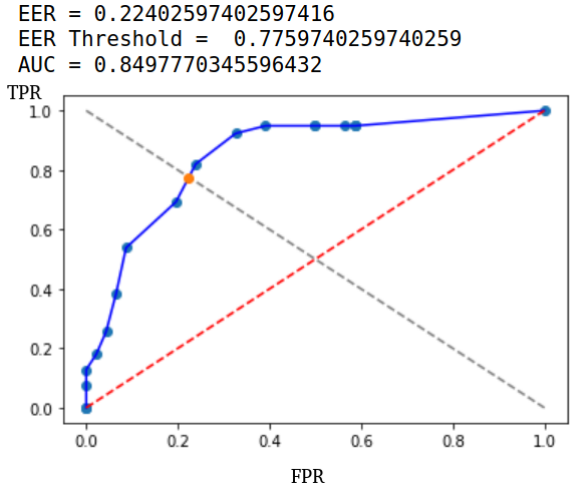


Figure 4.3: ROC for PED1, background subtraction based weighting



The results for UCSD PED1 data set is as shown in the table 4.1. According to the table, the proposed model outperforms all other models clearly in terms of AUC and EER, except the [46] model. The reason is due to several false positives present. But if we examine all the false positives carefully, many of them are not really false positives but true positives as the ground truth should be corrected to add those as well. Some examples are extensively discussed under the section, 'True Positives Outside Ground Truth'. When ground truth values are corrected to reflect the new ranges, the AUC becomes 91.6% and EER becomes 13.28%. The AUC is less than the state-of-the-art. But the EER is better than the state-of-the-art. Even though the numbers are not directly comparable due to changed ground truths, it indicates that the proposed model is more sensitive to anomalous events than the best performing models in the domain. The proposed model seem to perform more accurately than other models in general.
When the background subtraction based weighting is replaced with edge weighting the proposed system's AUC drops to 80.6% from 85% and it is less than the AUC by [21]. But still the EER is lower than the [21]. Hence the model performs at least as good as [21] for PED1 with edge based weighting. With the vanilla model as in 4.1, similar to the scenario of edge weighting, the model performs at least as good as [21] with AUC of 79.7 and a comparatively less EER. This EER is even lower than the edge based weighting case.

With background subtraction based weighting the model outperforms all other models, except [46] but including [21] with a clear gap of AUC(85%) and EER(22.4%). The weighting was done such that the foreground pixel values are weighted by 1

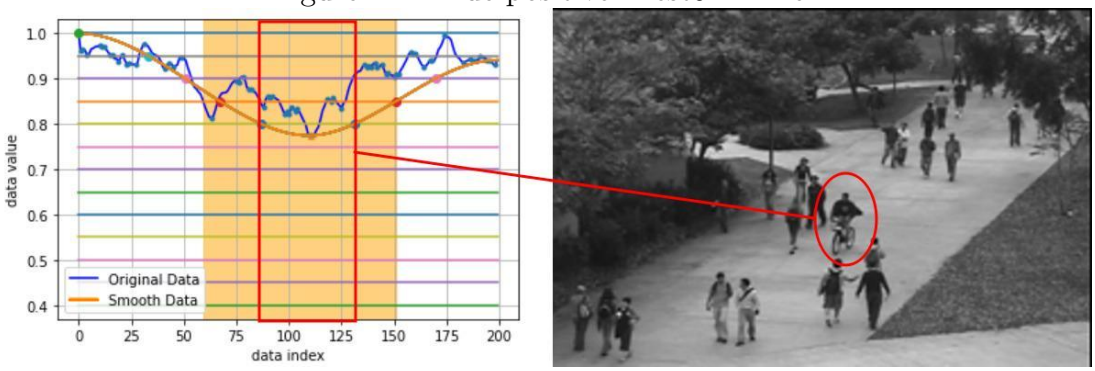
and the background is weighted by a value close to zero.

Under the following subsections, some examples of true positive, false positives, false negatives and ground truth corrected scenarios are described. All the examples are generated using the edge based weighting mechanism. The reason was that, when generated with edge based weighting the generated graph closely resembles to the vanilla graph but with higher differences between the event scores so that the graph scales vertically and the explaination is more clear. With background subtraction weighting, the shape of the graph has significant changes, hence not used for illustrations. Later in this sub section, all three model's outputs are compared altogether and there, some illustrations are used to show the improvements that the background subtraction brings.

4.3.1.1 True Positives

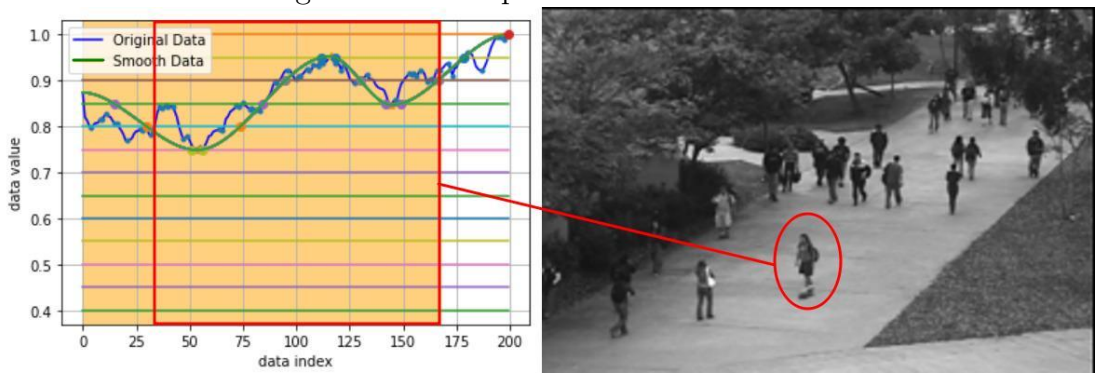
The below are some examples of true positives on various categories of anomalies.

Figure 4.4: True positive: Test01: Biker



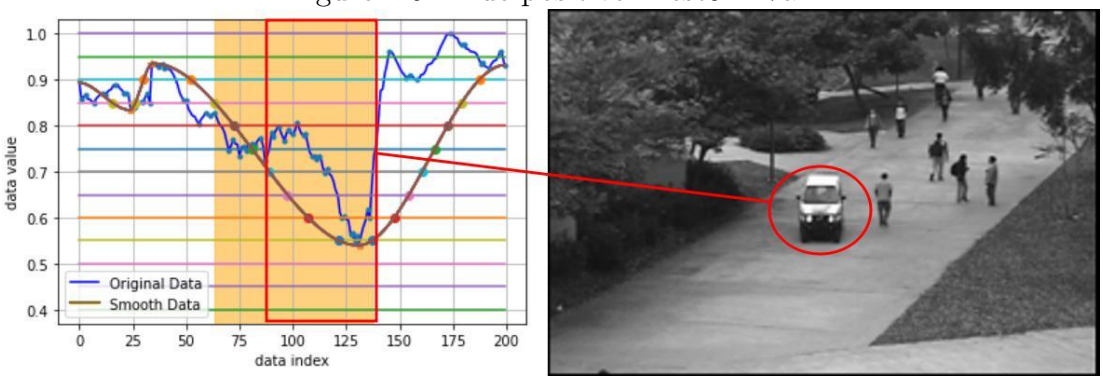
In the clip 4.4, a biker moves away from the camera and the model is able to successfully capture the event with a considerable difference in score.

Figure 4.5: True positive: Test04: Skater



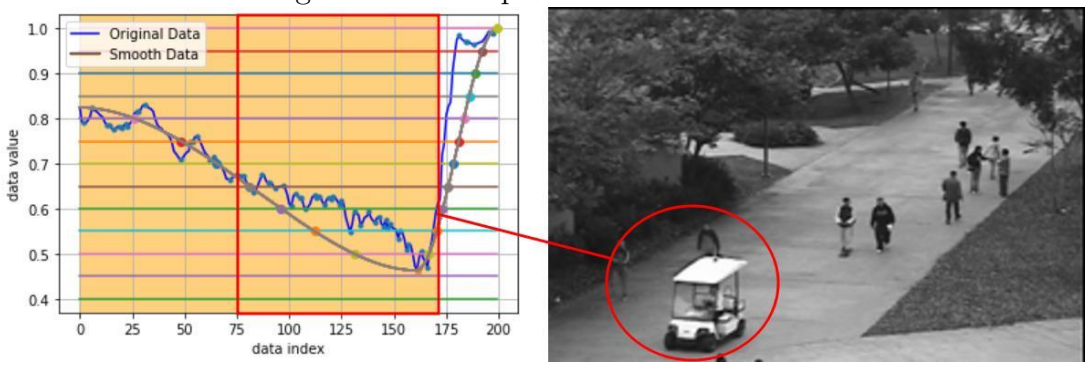
A skater in 4.5 moves among people away form the camera and the shape almost matches that of a normal walker. But the model is able to capture the anomaly based on the temporal behavior as the skater moves considerably faster than a normal walker.This temporal dimension is captured using the LSTM component in the model.

Figure 4.6: True positive: Test01: Van



This figure 4.6 shows a van moving towards the camera and this is easily captured even without the help of LSTM component as the holistic shape of the van is distinguishable from the shape alone with far apart scores.

Figure 4.7: True positive: Test01: Biker

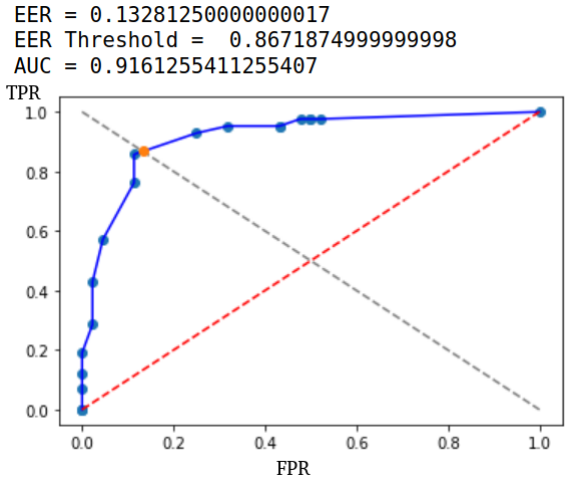


Similar to 4.6, this 4.7 also shows a vehicle, a cart that is moving towards the camera. This one is also detected in the same manner as the van.

4.3.1.2 True Positives Outside Ground Truth

Some events that were not included in the ground truth values were detected by the proposed model due to its high sensitivity. Some anomalies are obvious detections that are missed in the original ground truth values. But there are some others which are truly anomalies but not very significant in terms of the number of pixels.

Figure 4.8: ROC for PED1, ground truth correction



The new ROC curve after ground truth correction is as shown in 4.8. This shows a AUC of 91.6% and EER of 13.28%.

There are 11 such scenarios and 10 out of 11 were successfully detected by the proposed model. Some of the examples are illustrated in this sections.

Figure 4.9: Ground truth correction

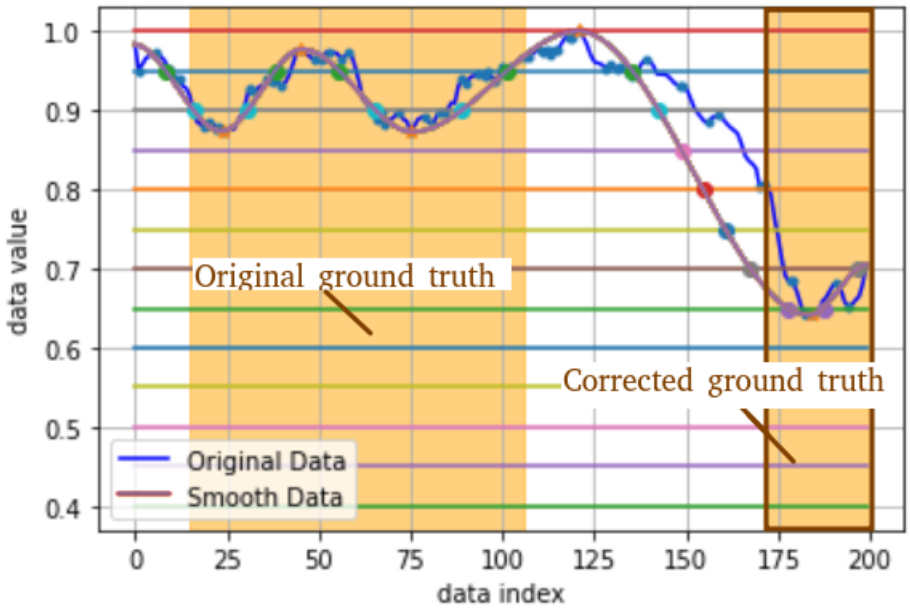


Figure 4.10: Test22: Corrected ground truth added in 4.9

In image 4.10 after the frame 172, a wheel chair appears from the far end of the road, but this is not included in the original ground truth values. Since the proposed model is very sensitive to such smaller details as the reconstruction is very accurate, such kind of detection turns out to be false positives compared with the original ground truth values.

Figure 4.11: Ground truth correction

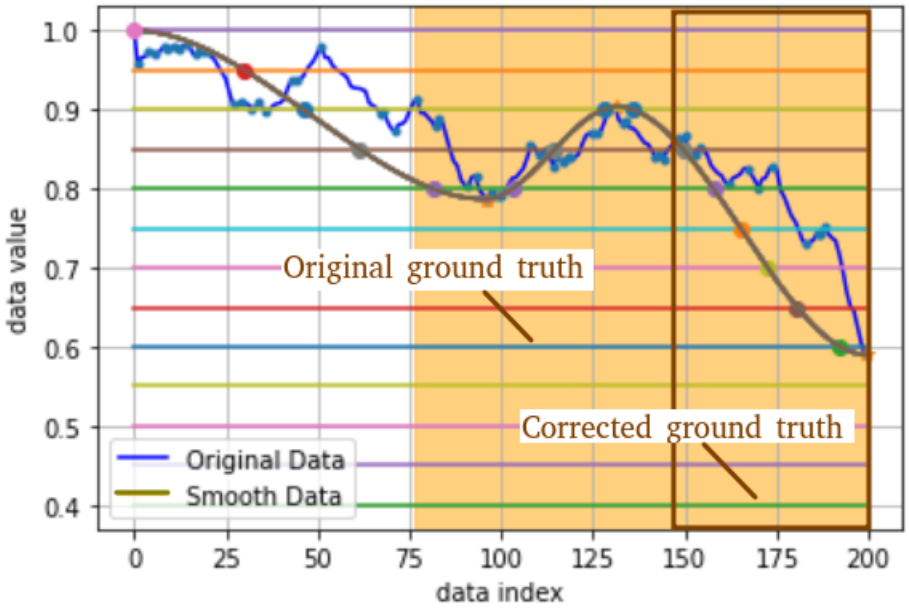


Figure 4.12: Test26: Corrected ground truth added in 4.11

In figure 4.12 the biker in the circled area is arriving after the frame 144. But the biker is travelling too slow to be detected as an anomaly. Hence not included in the original UCSD ground truths. The only useful information is the shape of the biker. This one is also detected as a false positive when compared with the original ground truths as shown in 4.11.

Figure 4.13: Ground truth correction

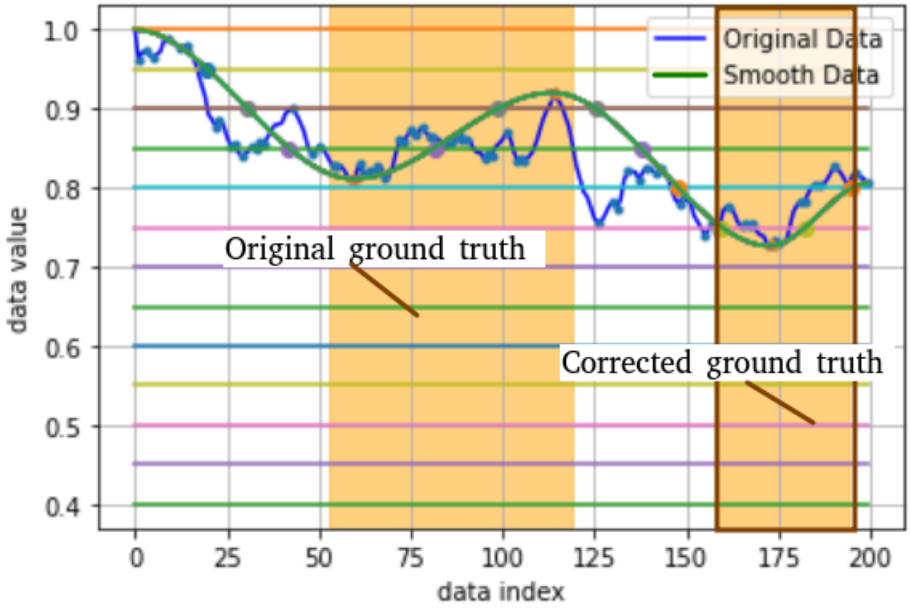


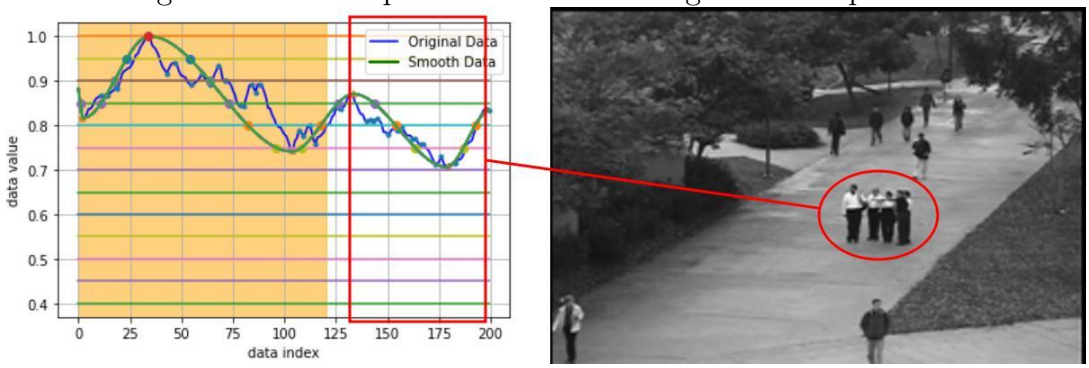
Figure 4.14: Test18: Corrected ground truth added in 4.13

In figure 4.14, a person is walking on the grass and as per the context, this can be considered as a restricted area. This scenario is an easy detection for the proposed models as the AAE's output is very accurate in the background areas compared to the foreground.

4.3.1.3 False Positives

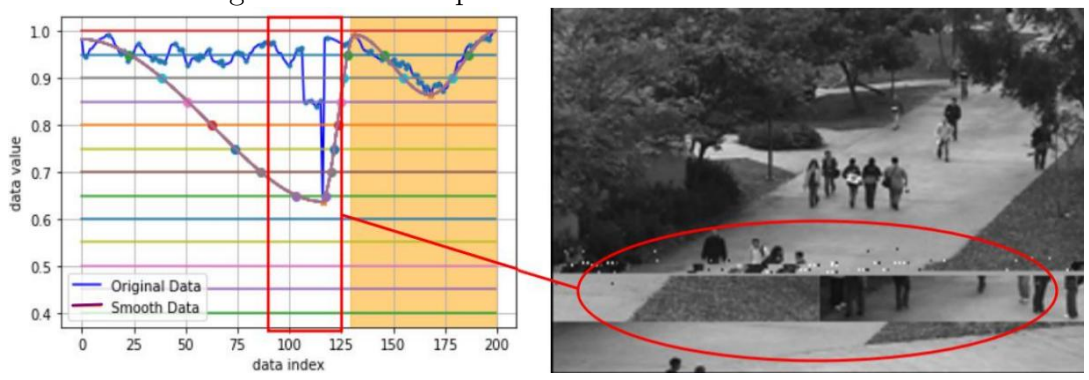
The proposed model has some false positives which are due to various reasons. Some are due to odd directions, and sometimes it is the bright clothes. It could be also due to change of the travel pattern and there is one false positive due to camera shake as well.

Figure 4.15: False positive: Test34: change of travel pattern



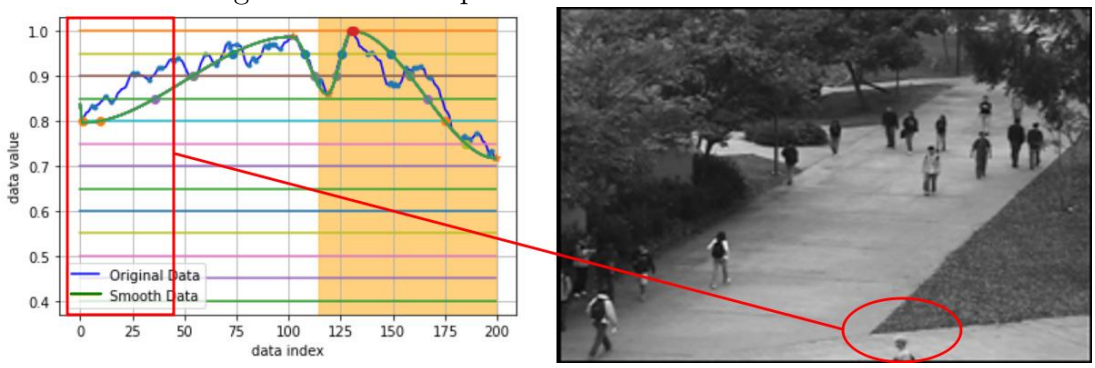
The group of people shown in 4.15 are traveling down the pathway and around the clip 130, they start slowing down and stop. This change of behavior triggers the model to wrongly predict their motion and the model to reproduce with higher error.

Figure 4.16: False positive: Test12: camera shake



There is one clip with camera shake. This affects the image predictions after a few frames and leads to higher error rate.

Figure 4.17: False positive: Test11: Odd direction

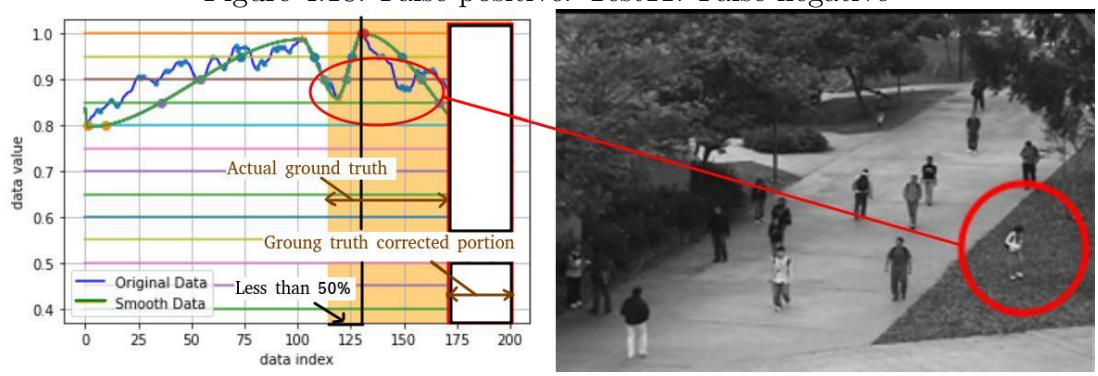


At the beginning of the clip, a person appears from the bottom right corner and walks towards left while his upper body becomes visible in some number of frames. This falls under odd direction category but since this is not considered as anomaly, this is categorized as false positive.

4.3.1.4 The False Negatives

There is one true false negative that is not detected even at the threshold value of 1.

Figure 4.18: False positive: Test11: False negative



As it is seen in 4.18, even though the natural graph has been able to capture the anomaly at the threshold of 0.9 with at least 50% coverage, the smoothed graph has not been able to capture one minima and the 50% rule does not comply. Hence categorized as false negative. The ground truth corrected portion is not considered for calculation.

4.3.2 UCSD PED2

Figure 4.19: ROC for PED2, vanilla model with uniform weighting

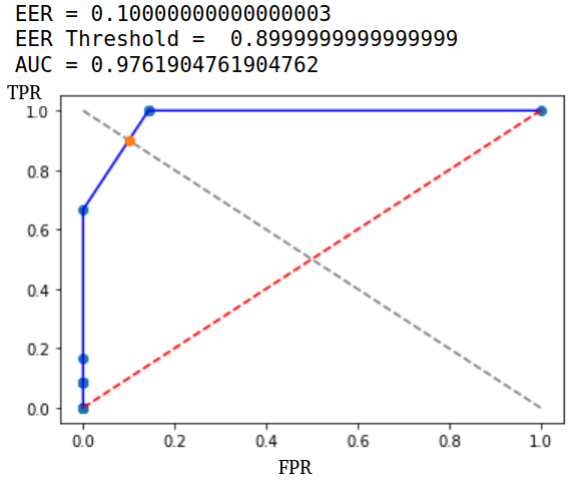


Figure 4.20: ROC for PED2, edge based weighting

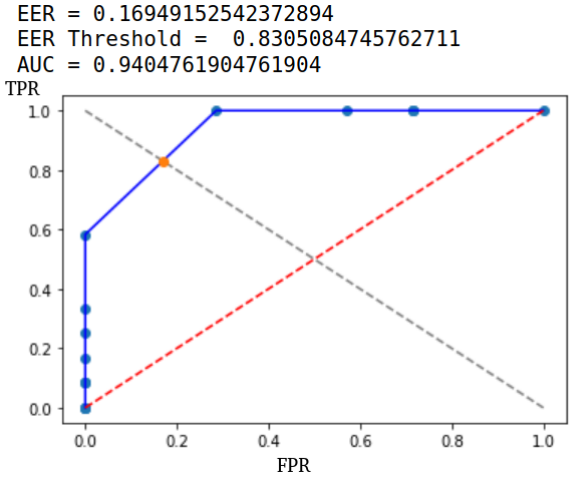
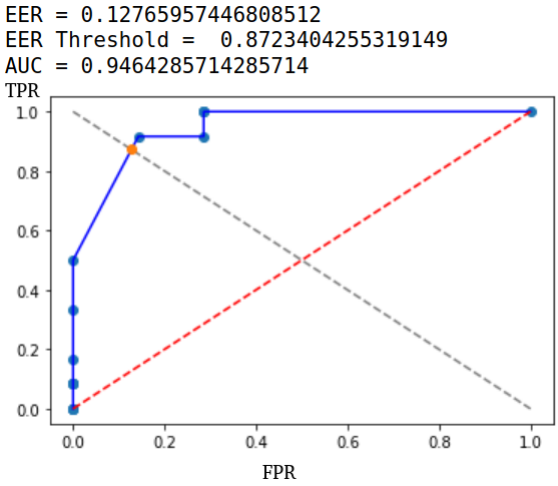


Figure 4.21: ROC for PED1, background subtraction based weighting



The performance metrics for PED2 is comparatively better than the same metrics for PED1 as it's size and variety is less complex than PED2. The model proposed in [19] is the model that performs best among other models. The AUC of [19] is less than that of the edge based method, even though EER is better than the proposed. But other two proposed models clearly outperform in every aspect compared with other models.
This data set contains pedestrians walking parallel to the camera and as anomalies, similar to PED1, vehicles, bikers, skaters are the types of anomalies found. In PED2 no ground truth correction was needed.

The background subtraction based weighting is giving outstanding results where not only it increases AUC by 0.4 points but also it decreases the EER by approximately 4 degrees(from 16.9 to 12.7), compared with edge based method. But the best performance is achieved by the vanilla model where it gives an exceptional AUC of 97.6 and EER of 10.0. The proposed methods are setting a new state-of-the-art AUC and EER for PED2 data set by breaking the current best performance record achieved by [19], which is AUC of 93.0% and EER of 13%.

The weighting was done such that the foreground pixel values are weighted by 1 and the background is weighted by 0.5. If the background pixels were assigned with near zero values, even though it improves the score differences, the model produces some false positives due to occlusions.

4.3.2.1 True Positives

These are some examples of true positives.

Figure 4.22: Test04, PED2 true positive

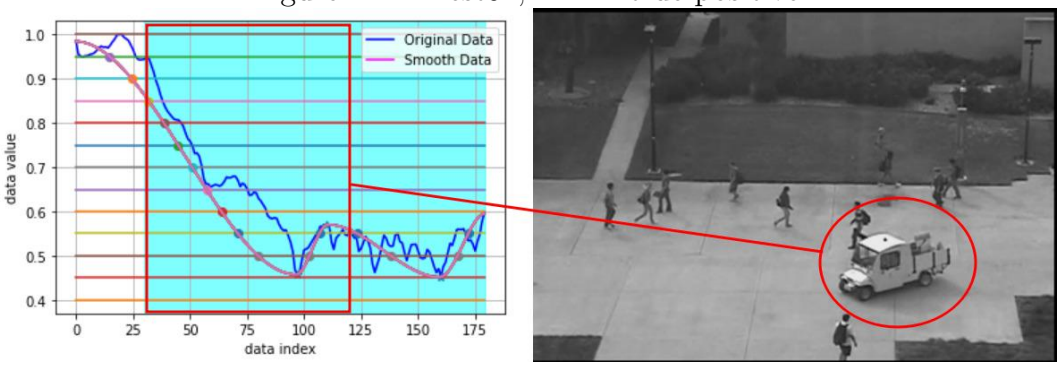


Figure 4.23: Test06, PED2 true positive

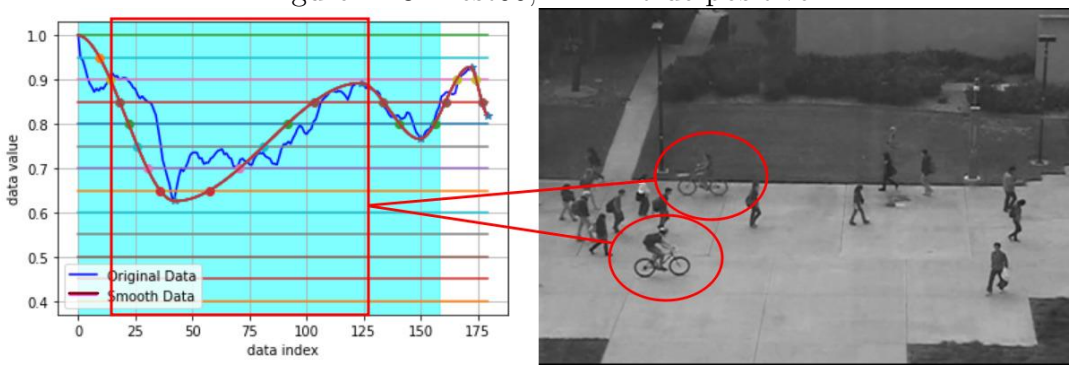
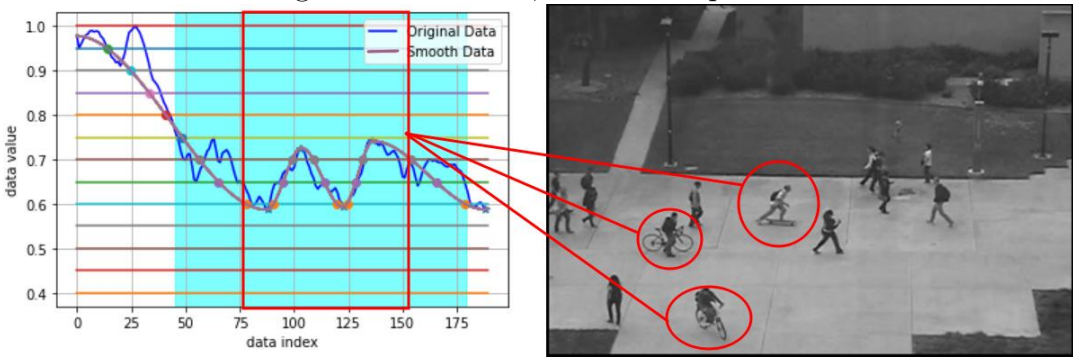


Figure 4.24: Test07, PED2 true positive



As shown in figures, 4.22, 4.23, 4.24, such events are easily distinguishable.

4.3.2.2 False Positives

The below false positive scenarios are having regular human activities but the scenes are crowded making some occlusions. The false positives below are due to different reasons like unusual objects in hand, suddenly stop moving etc.

Figure 4.25: Test01, Unusual objects in hand

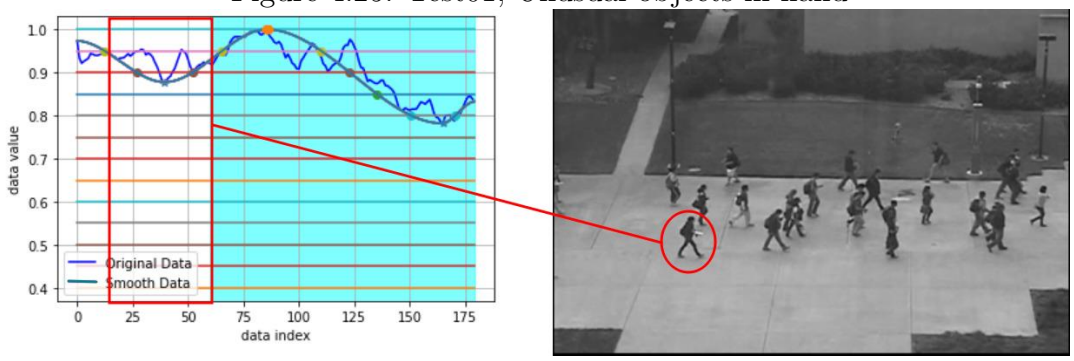


Figure 4.26: Test02, Suddenly stop moving

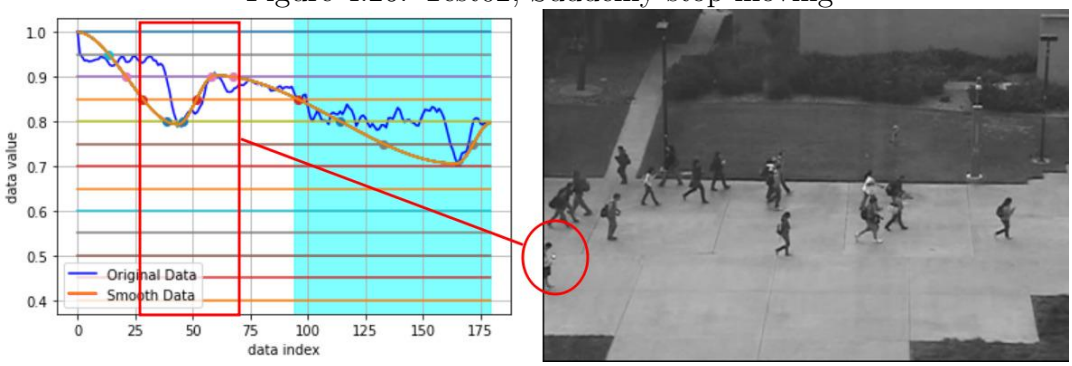
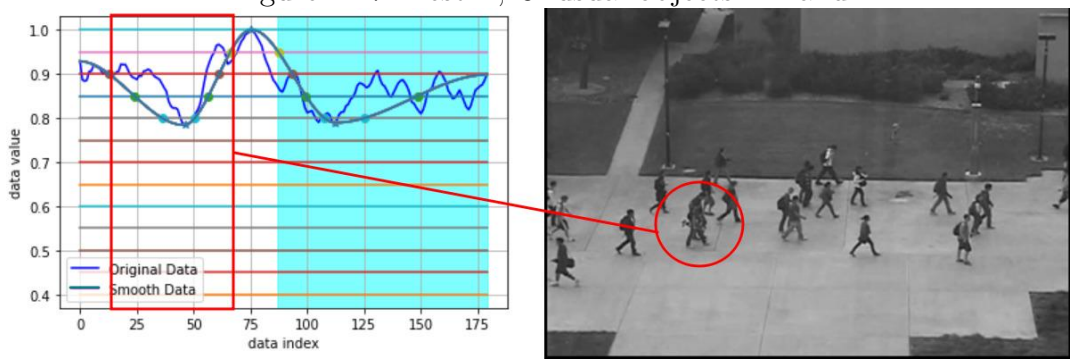


Figure 4.27: Test12, Unusual objects in hand



4.3.3 Improving Results - Weighted Pixel Values

In order to improve the scores, many methods were tried out. Among all the efforts weighting the most relevant pixels appropriately stood as a promising option. Hence, the option that was left was to find a suitable way to dynamically select the most important pixels and assign them a higher weight so that if a difference is found, it gets magnified to show a clear contrast in the scores. As a solution to find most important pixels in the image sequence, background subtraction and edge based weighting were used.
Using background subtraction is essentially as same as foreground segmentation. But in this research more attention was given to mobility of the objects in the frame. The Higher the mobility, the more it can be an anomalous event. This mobility factor serves the propose better than merely segmenting the foreground. But in contexts where anomalous events are not so mobile, this method may not be very effective. But it is not a common scenario. In such cases, a simple convolutional autoencoder without LSTM layers will suffice as there is less movement involved. But since the sole purpose of this research is to detect crowd anomalies in CCTV images, this assumption is always true.
This weighting is done in both reference and reconstructed images before calculating the similarity error. There was a significant difference in scores when background subtraction based weighting was introduced. The following sub sections illustrate how the scores get improved with weighting methodologies. PED2 data set is used to illustrate the difference.

4.3.3.1 Weighting Based Differences in Score

This sub section illustrates the anomaly score variations based on the methodology used to weight the pixels. According to the illustrations, the below key points can be observed,

- Vanilla model contains the basic shape that is detectable
- Edge based weighting always tends to improve the detectable scores but the area under detection does not seem to cover the majority
- Background subtraction based method gives significant improvement of the score value differences and also tends to maintain a good portion under the area of detection. The graph seems to be more flattened across the event range with a lower regularity score.

- 4.28 indicates that background subtraction without controlled weights may introduce some false positive detections.

Figure 4.28: Test002 : Left to Right in order: vanilla model, edge detected, background subtraction based, background subtraction based increased background weights

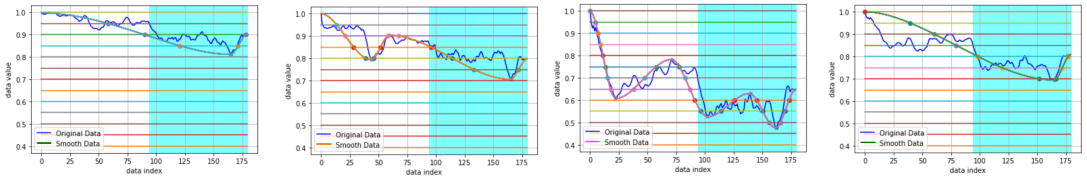


Figure 4.29: Test005 : Left to Right in order: vanilla model, edge detected, background subtraction based, background subtraction based increased background weights

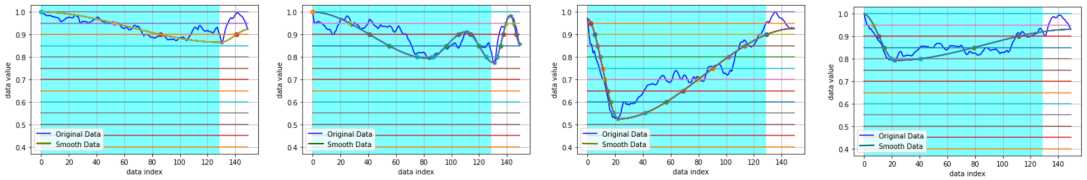
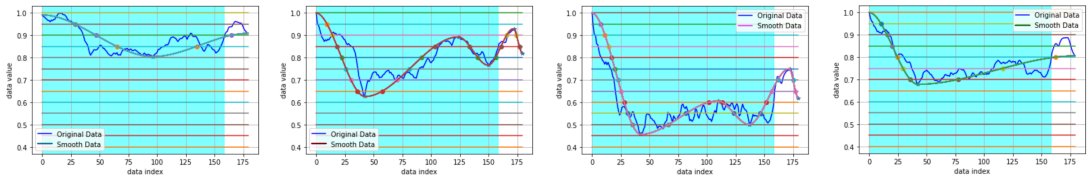


Figure 4.30: Test006 : Left to Right in order: vanilla model, edge detected, background subtraction based, background subtraction based increased background weights



4.3.4 Further Experiments

4.3.4.1 Avenue Data set

The Avenue data set contains 16 training videos and 21 testing videos that are of variable lengths. This data set has more foreground compared to UCSD data set.

Crowd movement is parallel to the camera in general. The anomalous events are of running and, and few other activities only. This does not have scenarios like vehicle movements, skaters, carts etc. This is not very widely cited as much as the UCSD data set. Hence was only experimented as part of robustness testing. The training was done for 400 epochs. Some examples of the detections obtained are shown below. These detections were performed by reducing the frame rate by half.

Figure 4.31: 04.avi : Running behavior

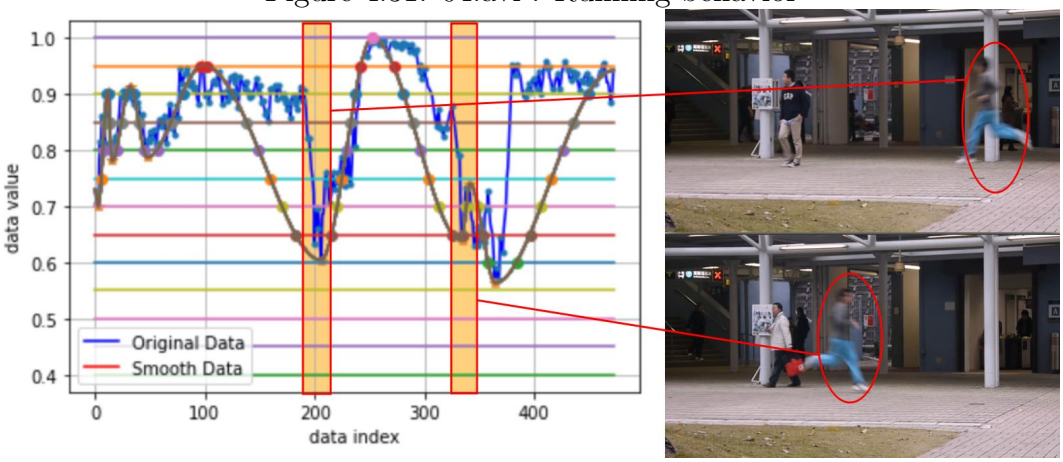


Figure 4.32: 05.avi : Throwing objects

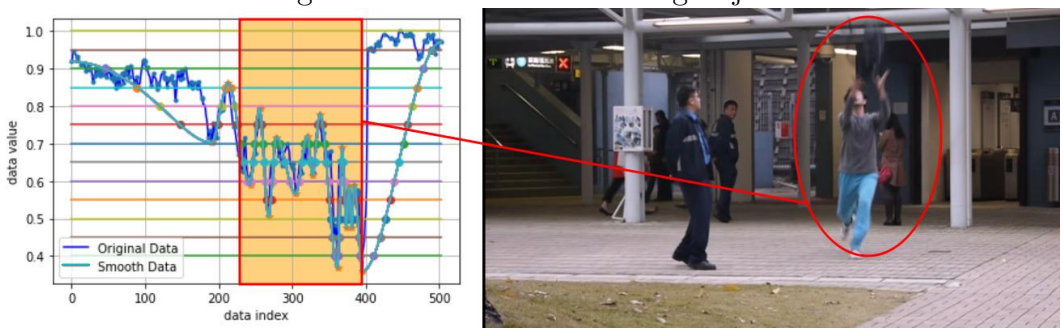
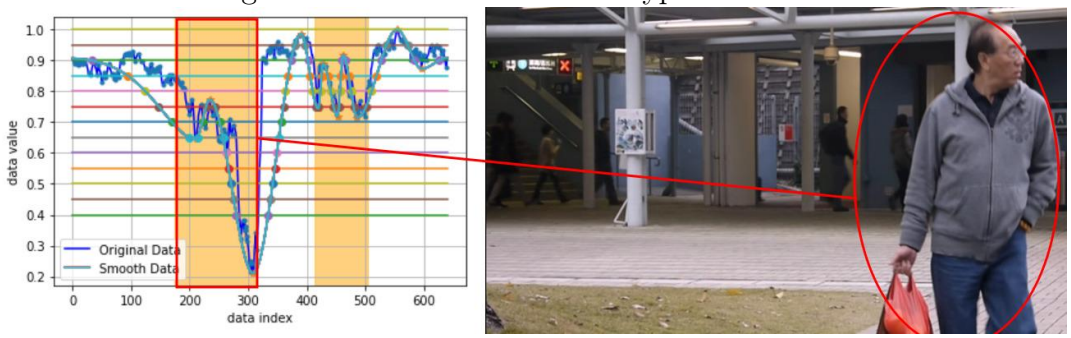


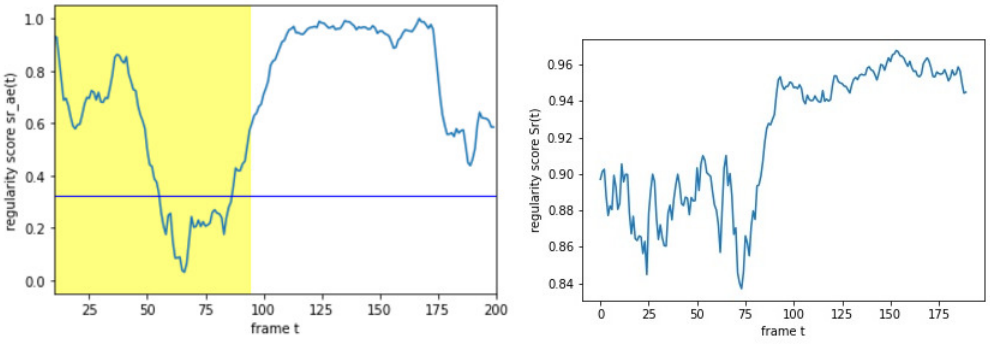
Figure 4.33: 06.avi : Different types of behavior



4.3.4.2 Latent Space Patterns

Since the model trained is a adversarial autoencoder, the latent space regularization is used to approximate a given distribution, standard normal distribution in this research, would reveal information related to anomalous behavior. Based on that intuition, the discriminator score during training was observed on a single test case. The below image is taken from the discriminator output at the 100^{th} epoch while training of PED1 data set.

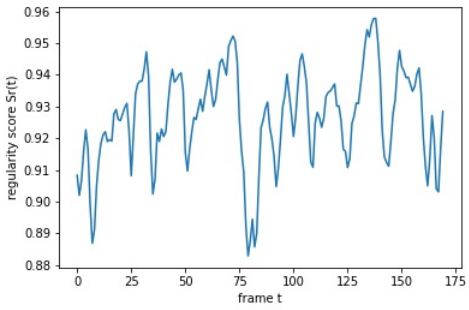
Figure 4.34: Test008 : PED1 — Left:- reconstruction error, Rigt:- discriminator score



The 4.34 clearly shows a pattern in the discriminator score where the same pattern is followed in the reconstruction error.

The below image 4.35 is obtained from the 100^{th} epoch while training PED2.

Figure 4.35: Test001 : PED2 — discriminator score



There is no clearly visible pattern observed in the 4.35.

As it seems, the information from the latent space indicates some pattern change for anomalous behaviors but this may need experimenting with different distributions that would enhance the pattern observability in the discriminator score.

4.4 Conclusion

The main focus of this research was to investigate and come up with an anomaly detection methodology that minimizes the supervising effort. The model that was created was an Adversarial Autoencoder based model, which requires no supervision from an exterior source, instead it self-supervises and converges to a solution.

Adversarial Autoencoders are regularized in the latent space and it greatly improves the quality of the output. The latent space information is also a good source of information which can be used to cluster the data. But in the domain of anomaly detection we do not have any information of anomalous clusters during training, due to which it is difficult to utilize in the manner of a clustering algorithm. Hence the reconstruction error is utilized to find the anomalous events.

In order to gain better outcomes, pixel based weights are applied. Background subtraction is used to remove the background and retain the moving objects which are used to weight individual pixel values with slightly higher values than the rest. The improvements were clearly visible in the AUC and EER values obtained against other models. The proposed model was even capable of setting new state-of-the-art values for PED2 in terms of AUC and EER.

In this research, three types of models(Vanilla model and two other variations) were proposed. Each of them were nothing but a derived version of the vanilla

model introduced. The performance of the three models were compared in terms of AUC and EER with other previous models introduced in the domain of crowd anomaly detection. All three types of the proposed models, outperform all other prevailing state-of-the-art models with exceptional scores, AUC/EER - PED1 - 85%/22.4%, and PED2 - 94.6%/12.8%. There is one exception with [46] where the proposed model falls a little behind on the AUC values obtained by [46]. But the ground truth correction analysis reveals that, the proposed model's sensitivity is better than any other model in the domain including [46].

4.5 Future Work

This research fills the first step of a long journey of creating a comprehensive system that can detect sophisticated anomalous events. This is currently detecting the unseen behavior as anomalies. But if this system is integrated with a behavioral analysis model and a face recognition model, this could also be used to detect suspecting behaviors as well. For example, if the same person moves around the same area or if he shows some repetitive behaviors & etc. In order to achieve this, we need to have another supporting system that can perform human behavioral analysis and face recognition based tracking. The systems proposed by [52] can be integrated for real time fast human pose recognition and the system proposed by [53] can be used for human face recognition based tracking. There is also room for a system that can analyze non-human objects that may cause suspicion. All such complex integrations open doors for a vast set of capabilities that this system could be developed into.
The background subtraction performs a great job in terms of making a contrast in the regular and abnormal events in the output score. This can be improved by better background removal techniques. For instance a variational autoencoder would do a better job in background seperation. Another idea worth experimenting would be to use a foreground segmentation mechanism to weight the local regions. This may also be done with an architecture similar to the famous U-Net.

In [44] they have used a adversarial dual autoencoders upon MNIST data in which two identical autoencoders train adversarially to win over each other. In a similar fashion, the loss function can be replaced with a suitable discriminator. This type of training tends to provide better results as the loss function is not fixed.

Another improvement would be to use the latent information to come up with an optimized anomaly score. [22] is using a similar mechanism but it needs lot of

smoothing in the output graph and may show a lot of inconsistency in different a real world scenario.

An obvious improvement would be to increase the depth and number of filters to capture more data. But this approach would cost in terms of processing power as well as time consumed for training. The latent variable width should not be increased unnecessarily as sparse latent spaces would always make the model over-fit and fail in training.

The next step of the research is to develop a anomaly localization technique. This is commonly done using optical flow based segmentation. But there can be many other novel ways of segmenting the anomaly detected in output score. The most basic way is to localize based on the score difference obtained for local regions. But this would need a lot of other steps to get a meaningful output.

The proposed system involves many new techniques to improve scores and these methodologies can be replaced with better approaches in the future that yield finer outcomes. This would require more experimenting on the techniques. Also another area to explore is to improve on the network layers and fine tune the model outcome, thus lot more room to explore in the domain.

Autoencoders is the most feasible way of performing state-of-the-art anomaly detection tasks. Recently generative adversarial networks have also been appearing in the literature, but they have their inherent problem of instability which hinders the performance when it comes to high dimensional data. But autoencoders on the other hand are more stable in nature and requires less parameter tuning compared to generative adversarial methods. Among all types of autoencoders variational autoencoders and adversarial autoencoders are more optimized for better outcomes. Thus, this methodology of using adversarial autoencoders for anomaly detection would be subjected to more improvements in the future without doubt.

Bibliography

[1] A. R. Forkan, I. Khalil, Z. Tari, S. Foufou, and A. Bouras, "A context-aware approach for long-term behavioural change detection and abnormality prediction in ambient assisted living," *Pattern Recognition*, vol. 48, no. 3, pp. 628–641, 2015.

[2] X. Wang, X. Ma, and E. Grimson, "Unsupervised activity perception by hierarchical bayesian models," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2007. [Online]. Available: https://doi.org/10.1109/cvpr.2007.383072

[3] J. Li, S. Gong, and T. Xiang, "Global behaviour inference using probabilistic latent semantic analysis," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2008, pp. 20.1–20.10, doi:10.5244/C.22.20.

[4] J. Kim and K. Grauman, "Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2009. [Online]. Available: https://doi.org/10.1109/cvpr.2009.5206569

[5] S. Basu, M. Bilenko, and R. J. Mooney, "A probabilistic framework for semi-supervised clustering," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '04. New York, NY, USA: ACM, 2004, pp. 59–68. [Online]. Available: http://doi.acm.org/10.1145/1014052.1014062

[6] B. Chandra and M. Gupta, "A novel approach for distance-based semi-supervised clustering using functional link neural network," *Soft Computing*, vol. 17, no. 3, pp. 369–379, aug 2012. [Online]. Available: https://doi.org/10.1007/s00500-012-0912-7

[7] R. Perdisci, G. Giacinto, and F. Roli, "Alarm clustering for intrusion detection systems in computer networks," *Engineering Applications of Artificial Intelligence*, vol. 19, no. 4, pp. 429–438, jun 2006. [Online]. Available: https://doi.org/10.1016/j.engappai.2006.01.003

[8] S. Calderara, R. Cucchiara, and A. Prati, "Detection of abnormal behaviors using a mixture of von mises distributions," in *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*. IEEE, sep 2007. [Online]. Available: https://doi.org/10.1109/avss.2007.4425300

[9] L. Ballan, M. Bertini, A. D. Bimbo, L. Seidenari, and G. Serra, "Effective codebooks for human action categorization," in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*. IEEE, sep 2009. [Online]. Available: https://doi.org/10.1109/iccvw.2009.5457658

[10] W. Kejun and P. P. Oluwatoyin, "Ant-based clustering of visual-words for unsupervised human action recognition," in *2010 Second World Congress on Nature and Biologically Inspired Computing (NaBIC)*. IEEE, dec 2010. [Online]. Available: https://doi.org/10.1109/nabic.2010.5716377

[11] J. D. Banfield and A. E. Raftery, "Model-based gaussian and non-gaussian clustering," *Biometrics*, vol. 49, no. 3, p. 803, sep 1993. [Online]. Available: https://doi.org/10.2307/2532201

[12] I. Tziakos, A. Cavallaro, and L.-Q. Xu, "Event monitoring via local motion abnormality detection in non-linear subspace," *Neurocomput.*, vol. 73, no. 10-12, pp. 1881–1891, Jun. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2009.10.028

[13] G. Malsiner-Walli, S. Frühwirth-Schnatter, and B. Grün, "Model-based clustering based on sparse finite gaussian mixtures," *Statistics and Computing*, vol. 26, no. 1-2, pp. 303–324, aug 2014. [Online]. Available: https://doi.org/10.1007/s11222-014-9500-2

[14] P. Bouttefroy, A. Bouzerdoum, S. Phung, and A. Beghdadi, "Local estimation of displacement density for abnormal behavior detection," in *2008 IEEE Workshop on Machine Learning for Signal Processing*. IEEE, oct 2008. [Online]. Available: https://doi.org/10.1109/mlsp.2008.4685511

[15] C. Clavel, L. Devillers, G. Richard, I. Vasilescu, and T. Ehrette, "Detection and analysis of abnormal situations through fear-type acoustic manifestations," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP 07*. IEEE, apr 2007. [Online]. Available: https://doi.org/10.1109/icassp.2007.367153

[16] M. Bahrololum and M. Khaleghi, "Anomaly intrusion detection system using gaussian mixture model," in *2008 Third International Conference on Convergence and Hybrid Information Technology*. IEEE, nov 2008. [Online]. Available: https://doi.org/10.1109/iccit.2008.17

[17] E. Bigdeli, B. Raahemi, M. Mohammadi, and S. Matwin, "A fast noise resilient anomaly detection using GMM-based collective labelling," in *2015 Science and Information Conference (SAI)*. IEEE, jul 2015. [Online]. Available: https://doi.org/10.1109/sai.2015.7237166

[18] D. Duque, H. Santos, and P. Cortez, "The OBSERVER: An intelligent and automated video surveillance system," in *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2006, pp. 898–909. [Online]. Available: https://doi.org/10.1007/11867586_81

[19] Y. S. Chong and Y. H. Tay, *Abnormal Event Detection in Videos Using Spatiotemporal Autoencoder*. Springer International Publishing, 2017, pp. 189–196. [Online]. Available: https://doi.org/10.1007/978-3-319-59081-3_23

[20] Medel, J. Ryan, Savakis, and Andreas, "Anomaly detection in video using predictive convolutional long short-term memory networks," Dec 2016. [Online]. Available: https://arxiv.org/abs/1612.00390

[21] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," *CoRR*, vol. abs/1604.04574, 2016. [Online]. Available: http://arxiv.org/abs/1604.04574

[22] A. Dimokranitou, "Adversarial autoencoders for anomalous event detection in images," 2017.

[23] M. Narasimhan and S. Kamath, "Dynamic video anomaly detection and localization using sparse denoising autoencoders," *Multimedia Tools and Applications*, vol. 77, pp. 13 173–13 195, 2018. [Online]. Available: https://doi.org/10.1007/s11042-017-4940-2

[24] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ser. ICML'11. USA: Omnipress, 2011, pp. 833–840. [Online]. Available: http://dl.acm.org/citation.cfm?id=3104482.3104587

[25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[26] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 1096–1103. [Online]. Available: http://doi.acm.org/10.1145/1390156.1390294

[27] D. Xu, Y. Yan, E. Ricci, and N. Sebe, "Detecting anomalous events in videos by learning deep representations of appearance and motion," *Computer Vision and Image Understanding*, vol. 156, pp. 117–127, mar 2017. [Online]. Available: https://doi.org/10.1016/j.cviu.2016.10.010

[28] H. Vu, T. D. Nguyen, A. Travers, S. Venkatesh, and D. Phung, "Energy-based localized anomaly detection in video surveillance," in *Advances in Knowledge Discovery and Data Mining*. Springer International Publishing, 2017, pp. 641–653. [Online]. Available: https://doi.org/10.1007/978-3-319-57454-7_50

[29] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," *CoRR*, vol. abs/1511.05440, 2015. [Online]. Available: http://arxiv.org/abs/1511.05440

[30] W. Lotter, G. Kreiman, and D. D. Cox, "Unsupervised learning of visual structure using predictive generative networks," *CoRR*, vol. abs/1511.06380, 2015. [Online]. Available: http://arxiv.org/abs/1511.06380

[31] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," *CoRR*, vol. abs/1506.04214, 2015. [Online]. Available: http://arxiv.org/abs/1506.04214

[32] J. R. Medel and A. E. Savakis, "Anomaly detection in video using predictive convolutional long short-term memory networks," *CoRR*, vol. abs/1612.00390, 2016. [Online]. Available: http://arxiv.org/abs/1612.00390

[33] ——, "Anomaly detection in video using predictive convolutional long short-term memory networks," *CoRR*, vol. abs/1612.00390, 2016. [Online]. Available: http://arxiv.org/abs/1612.00390

[34] L. Sun, K. Jia, T.-H. Chan, Y. Fang, G. Wang, and S. Yan, "DL-SFA: Deeply-learned slow feature analysis for action recognition," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2014. [Online]. Available: https://doi.org/10.1109/cvpr.2014.336

[35] Z. Zhang and D. Tao, "Slow feature analysis for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 436–450, mar 2012. [Online]. Available: https://doi.org/10.1109/tpami.2011.157

[36] V. R. Kompella, M. Luciw, and J. Schmidhuber, "Incremental slow feature analysis: Adaptive low-complexity slow feature updating from high-dimensional input streams," *Neural Computation*, vol. 24, no. 11, pp. 2994–3024, nov 2012. [Online]. Available: https://doi.org/10.1162/neco_a_00344

[37] X. Hu, Y. Huang, H. Zhang, H. Wu, and S. Hu, "Video anomaly detection using deep incremental slow feature analysis network," *IET Computer Vision*, vol. 10, no. 4, pp. 258–267, jun 2016. [Online]. Available: https://doi.org/10.1049/iet-cvi.2015.0271

[38] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, dec 2015. [Online]. Available: https://doi.org/10.1109/iccv.2015.510

[39] Y. Zhao, B. Deng, C. Shen, Y. Liu, H. Lu, and X.-S. Hua, "Spatio-temporal AutoEncoder for video anomaly detection," in *Proceedings of the 2017 ACM on Multimedia Conference - MM 17*. ACM Press, 2017. [Online]. Available: https://doi.org/10.1145/3123266.3123451

[40] A. Munawar, P. Vinayavekhin, and G. D. Magistris, "Spatio-temporal anomaly detection for industrial robots through prediction in unsupervised

feature space," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, mar 2017. [Online]. Available: https://doi.org/10.1109/wacv.2017.118

[41] D. D'Avino, D. Cozzolino, G. Poggi, and L. Verdoliva, "Autoencoder with recurrent neural networks for video forgery detection," *CoRR*, vol. abs/1708.08754, 2017. [Online]. Available: http://arxiv.org/abs/1708.08754

[42] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," 2015.

[43] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 2672–2680. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969033.2969125

[44] H. Son, Daisuke, Hashimoto, Kiyoshi, Kazuki, Sugiri, Shen, S. Mei, and Ueta, "Anomaly detection with adversarial dual autoencoders," Feb 2019. [Online]. Available: https://arxiv.org/abs/1902.06924v1

[45] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Lecture Notes in Computer Science*. Springer International Publishing, 2017, pp. 146–157. [Online]. Available: https://doi.org/10.1007/978-3-319-59050-9_12

[46] Xu, Dan, Ricci, Elisa, Yan, Yan, Sebe, and Nicu, "Learning deep representations of appearance and motion for anomalous event detection," Oct 2015. [Online]. Available: https://arxiv.org/abs/1510.01553

[47] Chen, Konukoglu, and Ender, "Unsupervised detection of lesions in brain mri using constrained adversarial auto-encoders," Jun 2018. [Online]. Available: https://arxiv.org/abs/1806.04972

[48] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 555–560, 2008.

[49] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 935–942.

[50] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1975–1981.

[51] T. Wang and H. Snoussi, "Histograms of optical flow orientation for visual abnormal events detection," in *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*, 2012, pp. 13–18.

[52] Bulat, Adrian, Kossaifi, Jean, Georgios, Pantic, and Maja, "Toward fast and accurate human pose estimation via soft-gated skip connections," Feb 2020. [Online]. Available: https://arxiv.org/abs/2002.11098

[53] Balaban and Stephen, "Deep learning and face recognition: the state of the art," Feb 2019. [Online]. Available: https://arxiv.org/abs/1902.03524

[54] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," vol. 2, 09 2004, pp. 28 – 31 Vol.2.

[55] "Persistence1d." [Online]. Available: https://ww2.mathworks.cn/matlabcentral/fileexchange/43540-persistence1d

Appendices

.1 Canny Edge Detection

Canny edge detector can suppress noise and detect edges at the same time. It is a multi step algorithm.

- Step 1: Smooth the image using a Gaussian kernel. This step helps reduce noise.

$$g(m,n) = G_\sigma(m,n) * f(m,n) \tag{1}$$

$$G_\sigma = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{m^2+n^2}{2\sigma^2}\right) \tag{2}$$

- Step 2: Computer the gradient of (1) using a gradient operator like Sobel, Roberts etc.

$$M(n,n) = \sqrt{g_m^2(m,n) + g_n^2(m,n)} \tag{3}$$

and

$$\theta(m,n) = \tan^{-1}\left[g_n(m,n)/g_m(m,n)\right] \tag{4}$$

- Step 3: Threshold value M:

$$M_T(m,n) = \begin{cases} M(m,n) & \text{if } M(m,n) > T \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

T is kept in a way such that all elements except the edges are suppressed.

- Step 4: In step 1 the edges might have been widened. Hence, suppress the non maxima pixels in the edges in M_T so as to make the edges thin.

- Use two different thresholds t_1 and t_2, such that $t_1 < t_2$ to come up with two binary images where in the image obtained by t_1 (image T_1)has more noise compared to the image obtained using t_2(image T_2). But on the other hand in the image obtained using t_2 has more accurate edges.

- Link the edge segments obtained in T_2 using the connectivity help from T_1 and complete the trace

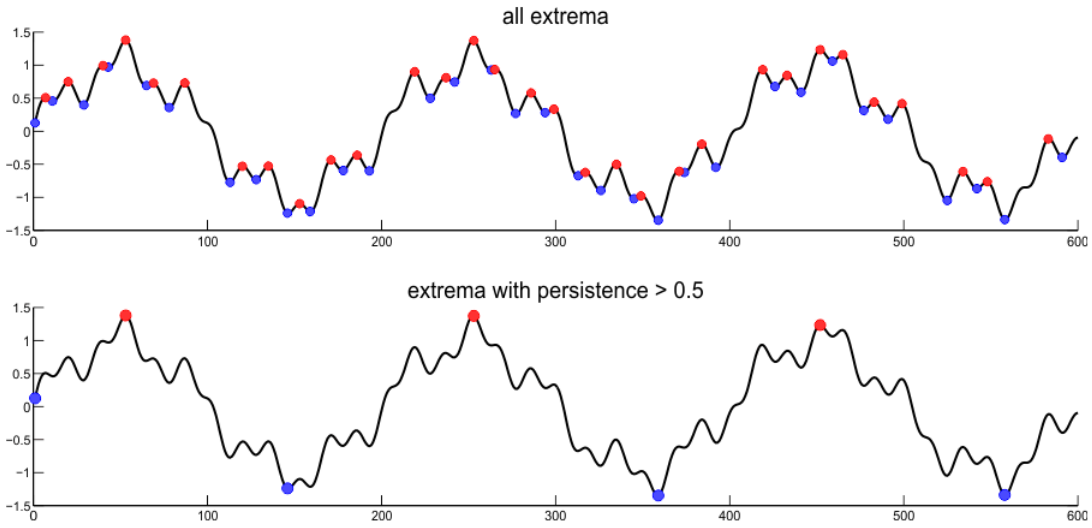
.2 MOG2 Background Subtraction

Background subtraction is the methodology of removing the background and retaining the foreground in a image sequence. In order to do this, a single image is not sufficient and a sequence of images is required. Among popular methods of finding the foreground, MOG and MOG2 are widely used. More details about MOG2 can be refered from [54]

.3 Persistence1d Algorithm

Persistence1D is a algorithm for locating local extremes and their persistence in one-dimensional data. Local minima and local maxima are extracted, compared and ranked in accordance with their constancy. More information can be referred from [55]

Figure 36: Maxima detection using persistence1d



Persistence1D detects all local minima and maxima. It pairs them according to persistence. Subsequent filtering reveals the most dominant local extrema of the function.

.4 UCSD Anomaly Detection Data set

More information can be found in *UCSD site*