RAMAIAH Institute of Technology

Department of Computer Science & Engineering

A Project Report on

Anomaly Detection in Videos Using Convolutional Autoencoder

Submitted in partial fulfillment of the requirements for the course

CS64: MINI PROJECT

By

1MS17CS071	Nandish Mahadev Karki
1MS17CS091	Rakshita N Patil
1MS17CS117	Somandra Singh Rathore
1MS17CS138	Yash Singh Chouhan

Under the guidance of

Mrs. Sini Anna Alex

Assistant Professor

Department of Computer Science and Engineering

MSRIT

M S RAMAIAH INSTITUTE OF TECHNOLOGY

(Autonomous Institute, Affiliated to VTU)

BANGALORE-560054

www.msrit.edu

2020

M. S. RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE – 560 054
(Autonomous Institute, Affiliated to VTU)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work titled "Anomaly Detection in Videos Using Convolutional Autoencoder" carried out by Nandish Mahadev Karki-1MS17CS071 , Rakshita N Patil-1MS17CS091 , Somandra Singh Rathore-1MS17CS117 and Yash Singh Chouhan-1MS17CS138 are bonafide students of M. S. Ramaiah Institute of Technology, Bengaluru, in partial fulfillment of the course Mini Project CS64 during the term Jan-May 2020. The project report has been approved as it satisfies the academic requirements for the aforesaid course. To the best of our understanding, the work submitted in this report is the original work of students.

Project Guide | **Head of the Department**

Mrs. SINI ANNA ALEX | **Dr. ANITA KANAVALLI**

External Examiners

Name of the Examiners: | **Signature with Date**

1.
2.

DECLARATION

We, hereby, declare that the entire work embodied in this mini project report has been carried out by us at M. S. Ramaiah Institute of Technology, Bengaluru, under the supervision of **Mrs**. **Sini Anna Alex, Assistant Professor,** Department of CSE. This report has not been submitted in part or full for the award of any diploma or degree of this or to any other university.

Signature
Somandra Singh Rathore
1MS17CS117

Signature
Yash Singh Chouhan
1MS17CS138

Signature
Rakshita N Patil
1MS17CS091

Signature
Nandish Mahadev Karki
1MS17CS071

ACKNOWLEDGEMENT

We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We would like to express our profound gratitude to the Management and **Dr. N.V.R Naidu** Principal, M.S.R.I.T, Bengaluru for providing us with the opportunity to explore our potential.

We extend our heartfelt gratitude to our beloved **Dr. Anita Kanavalli**, HOD, Computer Science and Engineering, for constant support and guidance.

We whole-heartedly thank our project guide **Mrs. Sini Anna Alex,** for providing us with the confidence and strength to overcome every obstacle at each step of the project and inspiring us to the best of our potential. We also thank her for her constant guidance, direction and insight during the project.

This work would not have been possible without the guidance and help of several individuals who in one way or another contributed their valuable assistance in preparation and completion of this study.

Finally, we would like to express sincere gratitude to all the teaching and non-teaching faculty of CSE Department, our beloved parents, seniors and my dear friends for their constant support during the course of work.

Somandra Singh Rathore

Yash Singh Chouhan

Rakshita N Patil

Nandish Mahadev Karki

ABSTRACT

In recent years there have been remarkable advances in areas such as machine learning and pattern recognition using convolutional neural networks (CNNs). Convolutional neural network (CNN) has considerable success in various machine learning tasks due to their powerful hierarchical feature learning ability in both spatial and temporal domains. We present an effective method for detecting anomalies in videos. We propose a spatiotemporal architecture for any abnormality detection in videos.

A Convolutional autoencoder- long short-term memory (LSTM) network is proposed to reconstruct raw data and perform anomaly detection based on reconstruction errors to resolve the existing challenges of anomaly detection in complicated definitions and background influence. Convolutional AEs and LSTMs are used to encode spatial and temporal variations of input frames, respectively. Comparisons with state-of-the-art approaches indicate the superiority of our approach in anomaly detection.

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

1.1 General Introduction

With the rapid growth of video data, there is an increasing need not only for recognition of objects and their behavior, but in particular for detecting the rare, interesting occurrences of unusual objects or suspicious behavior in the large body of ordinary data. Finding such abnormalities in videos is crucial for applications ranging from automatic quality control to visual surveillance. Meaningful events that are of interest in long video sequences, such as surveillance footage, often have an extremely low probability of occurring. As such, manually detecting such events, or anomalies, is a very meticulous job that often requires more manpower than is generally available. This has prompted the need for automated detection and segmentation of sequences of interest. However, present technology requires an enormous amount of configuration efforts on each video stream prior to the deployment of the video analysis process, even with that, those events are based on some predefined heuristics, which makes the detection model difficult to generalize to different surveillance scenes. Video data is challenging to represent and model due to its high dimensionality, noise, and a huge variety of events and interactions. Anomalies are also highly contextual, for example, running in a restaurant would be an anomaly, but running at a park would be normal. Moreover, the definition of anomaly can be ambiguous and often vaguely defined. A person may think walking around on a subway platform is normal, but some may think it should be flagged as an anomaly since it could be suspicious. These challenges have made it difficult for machine learning methods to identify video patterns that produce anomalies in real-world applications.

1.2 Problem Statement

There are many successful cases in the related field of action recognition. However, these methods only applicable to labelled video footages where events of interest are clearly defined and does not involve highly occluded scenes, such as crowded scenes. Furthermore, the cost of labelling every type of event is extremely high. Even so, it is not guaranteed to cover every past and future event.

The recorded video footage is likely not long enough to capture all types of activities, especially abnormal activities which rarely or never occurred.

Recent effort on detecting anomalies by treating the task as a binary classification problem (normal and abnormal) proved it being effective and accurate, but the practicality of such method is limited since footages of abnormal events are difficult to obtain due to its rarity. Therefore, many researchers have turned to models that can be trained using little to no supervision, including spatiotemporal features, dictionary learning and autoencoders. Unlike supervised methods, these methods only require unlabeled video footages which contain little or no abnormal event, which are easy to obtain in real-world applications.

1.3 Objectives of the project

• The main objective of this system is to detect and localize unusual objects, suspicious behaviors or irregular events in a scene. The method detects anomalous regions in a video and it is fast enough for real-time applications.

• The main abnormal behaviors that this project can detect are: Running, Motion in restricted areas.

1.4 Project deliverables

This is a project to represent video data by a set of general features, which are inferred automatically from a long video footage through a deep learning approach. Specifically, a deep neural network composed of a stack of convolutional autoencoders was used to process video frames in an unsupervised manner that captured spatial structures in the data, which, grouped together, compose the video representation. Then, this representation is fed into a stack of convolutional temporal autoencoders (CAE) to learn the regular temporal patterns

The method described here is based on the principle that when an abnormal event occurs, the most recent frames of video will be significantly different than the older frames. We train an end-to-end model that consists of a spatial feature extractor and a temporal encoder-decoder which together learns the temporal patterns of the input volume of frames.

The model is trained with video volumes consists of only normal scenes, with the objective to minimize the reconstruction error between the input video volume and the output video volume

reconstructed by the learned model. After the model is properly trained, normal video volume is expected to have low reconstruction error, whereas video volume consisting of abnormal scenes is expected to have high reconstruction error.

By thresholding on the error produced by each testing input volumes, our system will be able to detect when an abnormal event occurs.

The main characteristics of our approach:

• We wish to reduce the labor-intensive effort in feature engineering that results in a representation of the data that can support effective machine learning. This can be done by replacing low-level handcrafted features with learned hierarchical features. With the help of autoencoders, we are able to find representative features by learning from data instead of forming suitable features based on our knowledge.

• We replace traditional sparse coding methods with autoencoders. Unlike existing methods, there is no separation between extracting feature representation of videos and learning a model of features. In addition, by having multiple layers of hidden units in autoencoder, hierarchical feature learning can be achieved.

1.5 Current Scope

In recent times, surveillance systems are gaining a lot of popularity. The government, various organizations, residential societies, etc., are using these systems to keep a check on various activities for safety and security purposes.

Surveillance systems that include video analytics analyze video footage in real-time and detect abnormal activities that could pose a threat to an organization's security. Essentially, video analytics technology helps security software learn what is normal so it can identify unusual, and potentially harmful, behavior that a human alone may miss.

Rather than depend on solely human monitoring, intelligent systems instead notify security teams of potential threats as they happen, helping businesses prevent break-ins or illegal activity, as well as increasing human accuracy.

1.6 Future Scope

Our future work will be to extend the auto encoding framework to other video applications. This can be extended to detect various other kind of abnormal event that are usually encountered. By doing this a system which can detect any abnormality will be build which can deployed in various environments just encouraging portability. The algorithm should be carefully analyzed for areas where parallel processing is possible and suitably the algorithm should be tweaked which can help in achieving better accuracy

CHAPTER 2

PROJECT ORGANIZATION

2.1 Software Process Model

Process Model: Iterative Process Model

We selected the iterative process model as iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which can then be reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software for each cycle of the model

Our model comprises repeating the below-mentioned stages as a sequence. These are

Problem Statement Identification: the first step is where we decided the problem statement by understanding the problems in the modern world and discussing it with our respective guides. we went through a number of research papers to get better understanding of the problem and different approaches for their implementation.

Planning & Requirements: This stage is to map out the specification documents, establish software or hardware requirements, and generally prepare for the upcoming stages of the cycle and doing appropriate literature survey.

Analysis & Design: Once planning is complete, an analysis is performed to nail down the appropriate business logic, database models, and the like that will be required at this stage in the project. The design stage also occurs here, establishing any technical requirements (languages, data layers, services, etc.) that will be utilized in order to meet the needs of the analysis stage.

Implementation: With the planning and analysis out of the way, the actual implementation and coding process can now begin. All planning, specification, and design docs up to this point are coded and implemented into this initial iteration of the project.

Testing and Deployment: The next step is to go through a series of testing procedures to identify and locate any potential bugs or issues that have cropped up. After completing all the phases, software is deployed to its work environment

The biggest advantage of this model is that, it is implemented during the earlier stages of software development process, which allows developers and testers to find functional or design related flaws as early as possible, which further allows them to take corrective measures.

2.2 Roles and Responsibilities

Name	Roles	Responsibilities
Nandish Mahadev Karki	Data Collection and Workbook maintenance	1. Analyzing and preprocessing the dataset. 2. Updating workbook along with proper details.
Rakshita N Patil	Anomaly Detection, review and refinement and documentation	1. To determine whether a video frame is normal or anomalous. 2. Review and refinement of code. 3. Documentation of performed work.
Somandra Singh Rathore	Implementation of base model, Feature Learning and documentation	1. Features learning. 2. Implementation of neural network algorithm RNN, ConvLSTM. 3. Implementing base model. 4. Documentation of performed work.

Yash Singh Chouhan	Performing Training and Testing on the dataset and improving Accuracy, Documentation	1. Implementing improved model. 2. Performing training and testing on dataset. 3. Improving Accuracy. 4. Documenting the performed work.

CHAPTER 3

LITERATURE SURVEY

3.1 Introduction

Crowd analysis and scene understanding has drawn a lot of attention recently because it has a broad range of applications in video surveillance. Since crowd scenes have a large number of people accumulated with frequent and heavy occlusions, many existing technologies of detection, tracking, and activity recognition, which are only applicable to sparse scenes, do not work well in crowded scenes. Therefore, a lot of new research works, especially targeting crowd scenes, have been done in the past years. They cover a broad range of topics, including crowd segmentation and detection, crowd tracking ,crowd attribute recognition , crowd behavior analysis , and abnormality detection in a crowd .

In recent years, abnormal event detection in video has attracted a lot of attention within the pc vision analysis community. However, human eye is liable to distraction or temporary state thanks to long hours of observation. Additionally, the connection between human operators and therefore the sizable number of cameras is disproportionate, creating it a lot of difficult to find and answer all abnormal events that occur on the scene. This motivates the requirement for an automatic abnormal event detection framework victimization pc vision technology. Associate degree abnormal event has no consistent definition, because it varies per the context. In general, it's defined as a happening that stands out from the conventional behavior inside a specific context

3.2 Related Works with the citation of the References

[1] In this paper, they present a survey on relevant visual surveillance related researches for anomaly detection in public places, focusing primarily on roads. Since the underlying building block of a typical anomaly detection is learning, they emphasize more on learning methods applied on video scenes. they then summarize the important contributions made during last six years on anomaly detection primarily focusing on features, underlying techniques, applied scenarios and types of anomalies using single static camera. They treat anomaly detection by taking data as the primary unit detailing the learning

techniques, features used in learning, approaches employed for anomaly detection, and applied scenarios for anomaly detection.

Anomaly detection is a sub-domain of behavior understanding from surveillance scenes. With the availability of video feeds from public places, there has been a surge in the research outputs on video analysis and anomaly detection. Typically, anomaly detection methods learn the normal behavior via training. In this survey, they mainly explore anomaly detection techniques used in road traffic scenarios focusing on entities such as vehicles, pedestrian, environment and their interactions.

They have noted that scope of the study should cover the nature of input data and their representations, feasibility of supervised learning, types of anomalies, suitability of the techniques in application contexts, anomaly detection outputs and evaluation criteria. A typical anomaly detection framework is presented in Fig. Usually, anomaly detection systems work by learning the normal data patterns to build a normal profile.

With the advancements in machine learning techniques and affordable hardware, computer vision-based behavior analysis, anomaly detection and anomaly prediction can leapfrog in the coming years. Deep learning-based hybrid frameworks can handle diverse traffic scenarios. This can also help to build fully automatic traffic analysis frameworks capable of reporting events of interest to the stakeholders.

[2] Most applications are in surveillance, video content retrieval, and human–computer interfaces. This paper presents not only an update extending previous related surveys, but also a focus on contextual abnormal human behavior detection especially in video surveillance applications.

The following can be inferred about this interesting field of research. The definition anomaly can have some degree of ambiguity within a domain of application. Visual behaviors are complex and have much variety in an unconstrained environment. The influence of noisy data, the choice and representation of low-level features, significantly influences the discriminative power of the classifier.

Video quality, shadows, occlusion, illumination, moving camera, and complex backgrounds are challenges especially with a single-camera view. «Abnormal» activity can be maliciously adapted to appear as «normal» by the human agents. The methods that are adopted for feature representation, learning, and definition of anomaly vary according to the peculiarities of the

context. In practice, it may be easy to provide sufficiently many samples of normal activities, whereas it is difficult, if not impossible, to provide all possible examples and types of unusual activity that can happen in the scene.
Because abnormal events are rare, many attempt modeling normal behavior than using abnormal events as baseline. Another added complexity is that unusual behavior may not necessarily be abnormal, but simply a new instance of a normal behavior type and definitions of anomalies and available datasets, reporting various contexts and definition of anomalies. We note that while significant success has been achieved in this domain of research, some more work needs to be done as indicated next.

The fewness of datasets that are suitable as benchmarks to train and test contextual anomaly detection poses a challenge to the community. This understandably is because of the rarity and almost infinite variety of abnormal behaviors in real life. Such datasets should enable researchers assess how well an anomaly-detection algorithm performs in two important tasks, namely anomaly detection, this frame contain an anomaly or not. Ground truths for each frame need to be provided in the dataset to evaluate performance on this task.

The other is anomaly localization, where is the anomaly taking place? It is important in any anomaly-detection system to not only do well in detecting the presence of anomaly in the scene but identify where it is taking place. Although the processing rates of many of the competing techniques are often not stated, we observed in this survey that most are yet to be deployable in real time despite good performance. New methods are needed to overcome these limitations as well as the video data preprocessing and feature extraction so that these systems can be useful in real-time scenarios.

[3] This paper presents associate approach for the detection and localization of abnormal events in pedestrian areas. The goal is to style a model to observe abnormal events in video sequences exploitation motion and look info. Motion info is described through the employment of the speed and acceleration of optical flow and also the look info is described by texture and optical flow gradient. Not like literature ways, our planned approach provides a general resolution to observe each international and native abnormal event. What is more, within the detection stage, we have a tendency to propose a classification by native regions

In universe, sometimes traditional event samples square measure predominant, and abnormal event samples have very little presence. Due to this, the detection of abnormal events becomes even a lot of difficult. During this paper, to handle this downside, they tend to use the classification methodology planned in. In, the classification stage uses the minimum distance.

The most plan of this classification methodology is to go looking trained pattern that square measure kind of like the incoming pattern. That is, if the incoming pattern is analogous enough to a number of the noted patterns, then it's thought of as a traditional pattern. Otherwise, it'll be thought of as Associate in Nursing abnormal event. However, the classification supported the minimum distance gifts issues of detection thanks to some videos present the angle distortion within the scene.

This will significantly affect the feature extraction strategies because the extracted options will vary in keeping with their depth within the scene. During this paper, to handle the matter of perspective distortion and to get higher results, they tend to propose a classification supported native regions. In this paper, they have a tendency to propose a replacement approach primarily based within the motion and look info for abnormal event detection in huddled scenes. Motion options are appropriate for police investigation abnormal events with high speed motion.

[4] This paper is a survey on different approaches for Human Activity recognition which has utmost significance in pervasive computing due to its many applications in real-life. Human-oriented problems such as security can be easily taken care of by detecting abnormal behavior. Accurate human activity recognition in real-time is challenging because human activities are complicated and extremely diverse in nature. The traditional Closed-circuit Television (CCTV) system requires to be monitored all the time by a human being, which is inefficient and costly. Therefore, there is a need for a system which can recognize human activity effectively in real-time. It is time-consuming to determine the activity from a surveillance video, due to its size, hence there is a need to compress the video using adaptive compression approaches. Adaptive video compression is a technique that compresses only those parts of the video in which there is least focus, and the rest is not compressed. The objective of the discussion is to be able to implement an automated anomalous human activity recognition system which uses surveillance video to capture the occurrence of an unusual event and caution the user in real-time. So, the paper has two parts that include adaptive video compression approach of the surveillance videos and providing that compressed video as the input to detect anomalous human activity.

After researching the several methods that exists for human activity recognition it is noticed that the best suitable methods will be the ones that follow the unsupervised learning model, taking into consideration the kind of datasets available. Datasets available in large amounts are never labeled and hence Unsupervised or Semi supervised Deep Learning models must be selected. However, one cannot depend on a single methodology for activity recognition because the same algorithm works differently on different datasets. One must also consider the recognition performance and accuracy. The result of activity recognition is time consuming due to the size of the video hence it is better to adaptively compress the video before passing it for feature extraction. However, it is not advisable to suggest one algorithm as the best for implementation as each of them have their own advantages and disadvantages. Moreover, this is a field seeing rapid growth and there is a need to learn, research and analyze the best suitable method for the implementation

[5] In this paper, an unsupervised dynamic sparse coding approach is proposed in for detecting unusual events in videos, the method is based on online sparse constructability of query signals from an atomically learned event dictionary, which forms a sparse coding base. Someone uses a sparse set of

dictionary patterns which trained from normal video data and sparse coding error as a measure for detect abnormal events. An unsupervised probabilistic framework to detect abnormal events based on non-parametric statistical notion of spatiotemporal locality was developed. In recent years, deep learning becomes more and more popular as it can reduce the labour intensive effort in aspect of feature abstraction.

Neural Network to predict the subsequent features was proposed. Introduces a patch-based anomaly detection framework based on the Autoencoder reconstruction error and sparsity constraints. In addition, the noise needs to be manually provided, and selecting what is the most effective noise model is a not easy matter in an unsupervised setting.

Proposes a Convolutional Autoencoder method to learn normal pattern from normal datasets. During the test stage, the reconstruction error is used to obtain the regularity score for evaluation of the abnormal event. The model only uses the spatial feature, and ignores the temporal information. How to abstract better spatiotemporal features of the motion and appearance is the key issue for video processing. Thirdly, we show the effectiveness of our model on abnormal detection task.

Model forces the learned representation of LSTM Encoder-Decoder to contain more meaningful data and enhance the extrapolate capability of decoder by increasing information flow. Experiment shows that their hybrid auto encoder performs competitively to the state-of-the-art anomaly detection methods in both qualitative and quantitative way.

[6] The method described here is based on the principle that when an abnormal event occurs, the most recent frames of video will be significantly different than the older frames. The model in this

paper is trained with video dataset consists of only normal scenes, with the objective to minimize the reconstruction error between the input video volume and the output video volume reconstructed by the learned model. After the model is properly trained, normal video volume is expected to have low reconstruction error, whereas video volume consisting of abnormal scenes is expected to have high reconstruction error.

In pre-processing phase, the task of this stage is to convert raw data to the aligned and acceptable input for the model. As the number of parameters in this model is large, large amount of training data is needed. Following s practice, we perform data augmentation in the temporal dimension to increase the size of the training dataset.

In Feature Learning phase, proposed a convolutional spatiotemporal autoencoder to learn the regular patterns in the training videos. Convolutional layers are well-known for its superb performance in object recognition, while LSTM model is widely used for sequence learning and time-series modelling and has proved its performance in applications such as speech translation and handwriting recognition. After this model is trained and tested with Convolutional LSTM and analyzed, on the basis of regularity score videos are classified in anomalous or normal videos.

In this research, they have successfully applied deep learning to the challenging video anomaly detection problem. They formulated anomaly detection as a spatiotemporal sequence outlier detection problem and applied a combination of spatial feature extractor and temporal sequencer ConvLSTM to tackle the problem. The advantage of our model is that it is semi-supervised the only ingredient required is a long video segment containing only normal events in a fixed view. One idea is to add a supervised module to the current system, which the supervised module works only on the video segments filtered by proposed method, then train a discriminative model to classify anomalies when enough video data has been acquired

[7] An extensive amount of literature has been devoted to the field of anomaly detection in videos, and researchers have published valuable survey articles on the literature. The vast majority of anomaly detection methods in videos engage a handcrafted feature extraction stage, followed by establishing a pattern model using videos containing only regular activities. Any activity that does not fit the learned model is considered as an anomaly. Previous studies in abnormal event detection in videos concentrated on handcrafted features and deep learning-based approaches.

In another study, unlike convolutional trajectory-based approaches, they combined the output of trajectory and pixel-based analysis. On the opposite, the success of the trajectory-based methods degrades in crowded scenes due to difficulty of detecting and tracking objects. Researchers have

recently proposed generative models of regular activity patterns with the consideration of the difficulty of finding abnormal activity patterns.

Generative adversarial networks and especially convolutional neural networks are commonly used to build generative models. Convolutional neural network was initially used for anomaly detection to obtain both spatial and temporal features in the study. However, CNNs were not built to learn

temporal features in mind and are not a natural fit for videos. Convolutional Autoencoder is a good alternative to CNNs.

Proposed an approach that benefits from both fully connected autoencoder and fully convolutional feed-forward autoencoder to learn temporal regularity. The approach benefits from either state-of-the-art motion features or obtained features from autoencoder. To learn spatiotemporal representations better, Med eland Savakis proposed a convolutional long-short term memory network that is capable of encoding a video sequence. Especially, autoencoder based methods have come to the fore due to the easiness of its application and relatively reduced processing time for real-time surveillance systems.

[8] In this paper, they tried two approaches:

1. Pure Model:
   In the decoder of ConvAE, we concatenate the output of each ConvLSTM layer, and sent it to the next deconvolutional operation. The other part employs LSTM encoder decoder with

   ConvLSTM unit to learn the temporal video motion feature for the learned spatial feature maps of every time step. In specific, there are three ConvLSTM layers.
2. The Hybrid Model:
   The hybrid model is proposed to learn the time-series data based on two factors. First, in terms of LSTM network, it can remember the later context and meanwhile forget the former context since «gate» mechanism. Second, since LSTM network is a neural network model based on time-series data, the traditional back-propagation algorithm is not applicable to optimize this

model, but back-propagation through time. The traditional convolutional neural network increases the network depth by stacking several convolutional layers, so as to improve the accuracy.

However, a problem, arising from the excessive layering of convolutional layers, is gradient vanishing. The back-propagation algorithm cannot effectively update the gradient of previous layers, resulting in the inability to update the corresponding parameters of previous layers. ResNet uses “shortcut connection” method to build deeper network which is consist of many basic residual blocks. DenseNet not only connects the upper and lower layers directly, but also across layers.

In a word, the input of each layer of DenseNet comes from the output of all previous layers. The kernel of these two models aims to create short paths from early layers to later layers by means of skip connection. Inspired by, this shortcut connection is utilized in our hybrid model. In this way, we not only provide interaction and integrate information by cross temporal channels but also make sure the next ConvLSTM unit accepts tensor with the same dimension.

Hybrid model not only alleviate the gradient vanishing problem, also strengthen feature propagation and encourage feature reuse in both spatial flow and temporal flow. It should be noted that our pure model is simpler than hybrid model and does not utilize temporal short connection.

Anomaly Detection Algorithm:

Once our model is trained, we use our predictive branch to detect abnormal events. We employ the regularity score to evaluate performance of the model. The reconstruction error of a pixel’s intensity value *I* at location *(x, y)* in frame *t* of the video sequence is defined as following:

$e(x, y, t) = \_I(x, y, t) - fW(I(x, y, t))||2$

Given the reconstruction errors of the pixels of a frame *t*, we compute the reconstruction error of a frame by summing up all the pixel-wise errors: $e(t) = \_(x,y)\ e(x, y, t)$.We compute the regularity scores(t) of a sequence by the reconstruction error as follows:

$s(t) = 1 – (e(t) − e(t)min / e(t)max)$

When we get the regularity score of a video, we can determine the frame is normal or abnormal by set thresholding. The threshold which is selected by experience determines how sensitive we wish the detection system behave. In addition, in order to know the number of abnormal events in the given video, we explore local minima's that are very noisy and not all are meaningful in the time-series of regularity score to detect abnormal events. We use the Persistence1D algorithm to identify meaningful local minima's and obtain the abnormal temporal regions. In this step, if the distance of two local minima's is less than 50 frames, they are identified to be a part of same abnormal event.

Our model forces the learned representation of LSTM encoder–decoder to contain more meaningful data and enhance the extrapolate capability of decoder by increasing information flow.

[9]:The work is divided into three steps: background removal, feature extraction and behavior recognition, and anomaly detection and localization. To locate and detect anomalies from videos, a general trend followed is to learn normal behaviors due to the frequent occurrence In unsupervised learning, samples with similar nature are collected together. It is widely used for anomaly detection as it forms group of normal behaviors. All the approaches proposed in the literature extract motion by optical flow, spatiotemporal, or trajectory features. Variants of these raw features are also implemented, but the relation of object size and its motion is not yet incorporated. Inspired from Physics, in this work, a term momentum is defined as the product of motion and size of object. Thus, momentum differentiates behavior of normal and abnormal objects at both spatial and temporal levels.

[10] To identify the suspicious behavior of objects and to simplify the task of analysis of foreground region, background removal is done. Two approach applied for background removal are background subtraction and background removal by threshold on magnitude and optic flow. It uses histogram of

magnitudes, statistics, positional and momentum features to identify the anomalies. Features are computed at the local level so that behaviors can be captured densely and localization becomes easier A generator network takes input as a frame and produces a corresponding optical-flow image. The

abnormality detection problem in crowded scenes. A second generator network is fed with a real optical-flow image and outputs an appearance reconstruction.

Generative Adversarial Networks are used, an emerging approach for training deep networks using only unsupervised data. During testing time, the trained networks are used to generate appearance and motion information.

In these papers we addressed the problem of abnormality detection in crowd videos. We proposed a generative deep learning method based on two conditional GANs. Since our GANs are trained using only normal data, they are not able to generate abnormal events. At testing time, a local difference between the real and the generated images is used to detect possible abnormalities. Experimental results on standard datasets show that our approach outperforms the state of the art with respect to both the frame-level and the pixel-level evaluation protocols. As future work we will investigate the use of Dynamic Images as an alternative to optical-flow in order to represent motion information collected from more than one frame.

[11]Anomaly detection, also named as outlier detection, refers to detecting patterns in a given data set that do not conform to an established normal behavior, which is applicable in a variety of applications, such as intrusion detection, fraud detection ,fault detection ,system health monitoring ,event detection in sensor networks, and detecting eco-system disturbances. Let us clarify the abnormal event detection firstly. Normalization Cut is used to discriminate the abnormal clusters from normal clusters. According to the definition above, the abnormal events can be identified as irregular events from normal ones.

Depending on the specific scene, the abnormal event detection can be classified into those in crowded scenes and uncrowded scenes. For uncrowded scenario, as the foreground objects can be extracted easily from the background, binary features based on background model are usually adopted, such as Normalized Cut clustering Due to the object template can be initialized in the uncrowded scene, They use frame-difference for object localization and then generate the object trajectories by tracking.

[12] In this paper, authors propose an anomaly detection algorithm using weakly labeled training videos. That is they only know the video-level labels, i.e. a video is normal or contains anomaly somewhere, but we do not know where. This is intriguing because large number of videos can be annotated by only assigning video-level labels. To formulate a weakly-supervised learning approach, they resort to multiple instance learning (MIL). Specifically, anomaly is learnt through a deep MIL framework by treating normal and anomalous surveillance videos as bags and short segments/clips of each video as instances in a bag. Based on training videos, they automatically learn an anomaly ranking model that predicts high anomaly scores for anomalous segments in a video. During testing, a long untrimmed video is divided into segments and fed into our deep network which assigns anomaly score for each video segment such that an anomaly can be detected.

MIL solution is proposed to anomaly detection by leveraging only weakly labeled training videos and also MIL ranking loss with sparsity and smoothness constraints for a deep learning network to learn anomaly scores for video segments.

A large-scale video anomaly detection dataset consisting of 1900 real-world surveillance videos of 13 different anomalous events and normal activities captured by surveillance cameras is introduced. It is by far the largest dataset with more than 15 times videos than existing anomaly datasets and has a total of 128 hours of videos.

Experimental results on this dataset show that proposed method achieves superior performance as compared to the state-of-the-art anomaly detection approaches.

[13] In this paper, the deep learning technique is used for abnormal event detection by extracting spatiotemporal features from video sequences. Saliency information (SI) of video frames is first extracted as the feature representation in the spatial domain. The frame is firstly divided into image

patches, and then the saliency value of each patch is determined according to the difference between one patch and all the other patches from the features of color, intensity, and orientation.

Optical flow (OF) is estimated as an important feature of video sequences in the temporal domain. To extract the accurate motion information, multi-scale histogram optical flow (MHOF) can be obtained through OF. MHOF and SI are combined into the spatiotemporal features of video frames.

Finally, a deep learning network, PCANet, is adopted to extract high-level features for abnormal event detection. For different video sequences, the suitable PCANet should be selected for abnormal event detection. With different sizes of filter in PCANet, the accuracy of abnormal event detection

might be different. PCANet is able to extract better features from complex scenes. This also conforms to the original intention of deep learning. Because PCANet is a neural network with no feedback and unsupervised learning, PCANet model is more sensitive to the number of training samples during abnormal event detection.

Experimental results show that the proposed abnormal event detection method can obtain much better performance than the existing ones on the public video database.

[14] In this paper, a novel unsupervised learning approach for a video anomaly detection system based on convolutional auto-encoder architectures is introduced.

. The anomalies that occur in outdoor scenes are focused, considering the challenging publicly available UCSD anomaly detection datasets. The variational auto-encoder is used to decouple the components of the feature from each receptive field as much as possible so that it is easier and more accurate to model the Gaussian for the features.

A new baseline of the anomaly detection and location framework using the variational auto-encoder to learn the discriminative feature representations of appearance and motion patterns is proposed. The feature representations of all receptive fields are extracted at one time and model a unique Gaussian for each feature. The variational auto-encoder is used to decouple the components of the feature from each receptive field as much as possible so that it is easier and more accurate to model the Gaussian for the features.

The fundamental advantage of approach followed by the authors of this approach is the use of a variational convolutional auto-encoder. On the one hand, this approach can extract features independent of the prior knowledge of hand-crafted features (the input of our detection system are raw pixels) and dispenses with any object-level analysis, like object detection and tracking.

On the other hand, process of cropping the input into patches are omitted by the convolution principle of the convolutional neural network, which makes this framework simple and clear. The effectiveness and robustness of the proposed approach is demonstrated, showing competitive performance to

existing methods. In fact, the approach presented for the anomaly detection system can be viewed as a baseline of using a variational auto-encoder to detect anomalies in surveillance

video. Further research directions will include jointing the input with richer temporal and contextual information and combining the feature extraction with the final anomaly decision. Besides, we can learn from the deep neural network frameworks for object detection and classification tasks to design more sophisticated frameworks, in order to represent the multiple patterns from the input video.

[15] In this paper, a novel online adaptive method based on combination of pretrained 3D residual network and online classifier were developed and implemented. It is able to detect abnormal events

and localize complex abnormal events in non-crowded and crowded scenes, prevent the marginalization of normal behavior that rarely occurs during the training phase and adapt to the appearance of new normal events in the testing phase.
In addition, this method does not require pretreatment methods such as tracking or background subtraction. This method is based on two main stages: Spatiotemporal feature extraction without any need of training, and the use of robust incremental classifier that prevents the redundancy of information in CCTV. It can also either be used online or offline. Authors have tested their proposed methodology on two main datasets using crowded (Ped2) and non-crowded scenes (CapSec). The results from the Ped2 dataset showed high performance in detection and localization for abnormal events based on The EERFL and EERPL.
According to experimental results this method outperforms all existing techniques present in the literature and used for Ped2. Besides, the fastness and the simplicity of this method allow us to use it for real-world application (case of CapSec). The results presented in this paper showed the effectiveness of using this framework in detecting abnormal events. This method is robust, takes into account rare normal events present in the training phase. Besides, it can be incorporated in online CCTV. Moreover, the method can be adapted so that human operators select false alarms to prevent its future appearance, which is suitable for dynamic environment. In this method, the localization of abnormal events is reflected as patches in the original image. In some cases, these patches may overflow on normal regions.

3.3 Conclusion of Survey

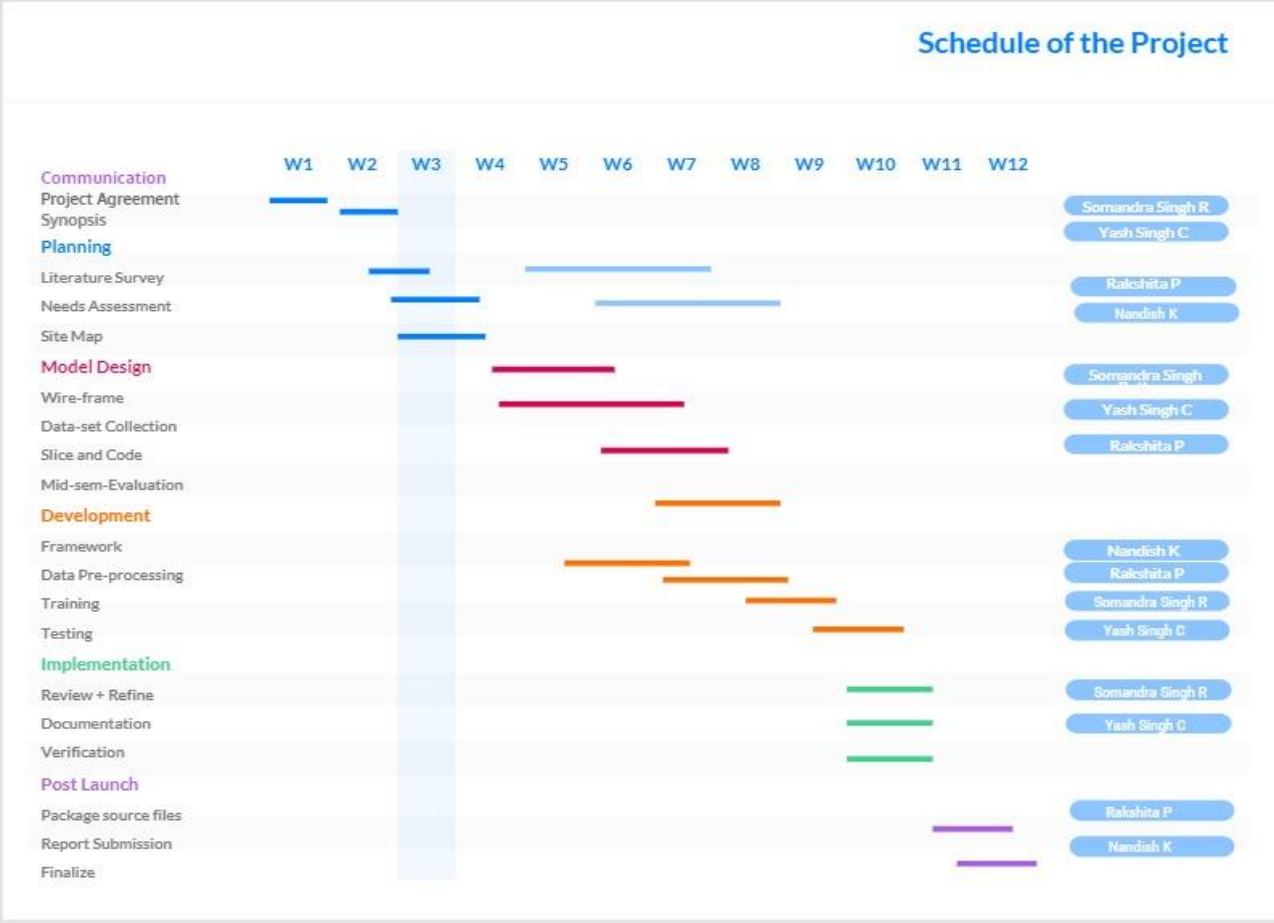
There are several methods for abnormal event detection according to the applications. First extraction of feature vector is done for moving objects. Then we can use probabilistic methods or frame-based classification methods. Some methods are rule based. At current, most of the methods consists of some steps: Feature computation, transform of the aggregated features to certain domains, building of model and perform detection.

Abnormal event detection in videos can be done using Autoencoders (Spatiotemporal, Variational, Convolutional), generative adversarial nets and Convo nets. Some deep learning techniques such as Gaussian Mixture Model, Multiple Instance Learning, can also be used.

CHAPTER 4

PROJECT MANAGEMENT PLAN

4.1 Schedule of the Project (Represent it using Gantt Chart)



4.2 Risk Identification

- First and very basic risk is lack of video data availability.
- Definition of Abnormal event is not defined an event which is abnormal for a scenario is normal for another scenario.
- Memory Optimization is also we have to take care of.

CHAPTER 5

SOFTWARE REQUIREMENT SPECIFICATIONS

5.1 Purpose

The purpose of this document is to build a software to detect anomalies(abnormal events) in videos using spatiotemporal encoder and to automate the quality of visual surveillance and enhance the safety of general public.

5.2 Project Scope

The combined objective of this project is to detect any anomalies during the video surveillance, monitor them and handle the anomalies. The method used to detect the anomalies is Spatiotemporal Autoencoder. During the course of time we select the training dataset and train the model. once we have trained the model sufficiently, we put the real instance and see whether it can detect any anomaly in the video that is fed.

5.3 Overall Description

5.3.1 Product perspectives

A novel framework to represent video data by a set of general features, which are inferred automatically from a long video footage through a deep learning approach. Specifically, a deep neural network composed of a stack of convolutional autoencoders was used to process video frames in an unsupervised manner that captured spatial structures in the data, which, grouped together, compose the video representation. Then, this representation is fed into a stack of convolutional temporal autoencoders to learn the regular temporal patterns.

5.3.2 Product Features

The method described here is based on the principle that when an abnormal event occurs, the most recent frames of video will be significantly different than the older frames. The model is trained

with video volumes consists of only normal scenes, with the objective to minimize the reconstruction error between the input video volume and the output video volume reconstructed by the learned model. After the model is properly trained, normal video volume is expected to have low reconstruction error, whereas video volume consisting of abnormal scenes is expected to have high reconstruction error. we train a end-to-end model that consists of a spatial feature extractor and a temporal encoder-decoder which together learns the temporal patterns of the input volume of frames.

1. Preprocessing: The task of this stage is to convert raw data to the aligned and acceptable input for the model. As the number of parameters in this model is large, large amount of training data is needed. Following s practice, we perform data augmentation in the temporal dimension to increase the size of the training dataset.
2. Feature Learning: We propose a convolutional spatiotemporal autoencoder to learn the regular patterns in the training videos. Our proposed architecture consists of two parts — spatial autoencoder for learning spatial structures of each video frame, and temporal encoder-decoder for learning temporal patterns of the encoded spatial structures
    - Autoencoder: Autoencoders, as the name suggests, consist of two stages: encoding and decoding. It was first used to reduce dimensionality by setting the number of encoder output units less than the input. With the activation function chosen to be nonlinear, an autoencoder can extract more useful features than some common linear transformation methods such as PCA.
    - Spatial Convolution: The primary purpose of convolution in case of a convolutional network is to extract features from the input image. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data. A convolutional network learns the values of these filters on its own during the training process, although we still need to specify parameters such as the number of filters, filter size, the number of layers before training.

- Recurrent Neural Network (RNN): RNN works just like a feedforward network, except that the values of its output vector are influenced not only by the input vector but also on the entire history of inputs.
- Long Short-Term Memory (LSTM): long short-term memory (LSTM) model which incorporates a recurrent gate called forget gate. With the new structure, LSTMs prevent

  backpropagated errors from vanishing or exploding, thus can work on long sequences and they can be stacked together to capture higher level information
- Convolutional LSTM: ConvLSTM has its matrix operations replaced with convolutions. By using convolution for both input-to-hidden and hidden-to-hidden connections, ConvLSTM requires fewer weights and yield better spatial feature maps

3. Regularity Score: Once the model is trained, we can evaluate our models performance by feeding in testing data and check whether it is capable of detecting abnormal events while keeping false alarm rate low. we used the formula to calculate the regularity score for all frames, the only difference being the learned model is of a different kind.
4. Anomaly Detection:
   - Thresholding: The threshold determines how sensitive we wish the detection system to behave — for example, setting a low threshold makes the system become sensitive to the happenings in the scene, where more alarms would be triggered.
   - Event count: To reduce the noisy and unmeaningful minima in the regularity score, we applied Persistence1D algorithm to group local minima with a fixed temporal window of 50 frames. We assume local minima within 50 frames belong to the same abnormal event.

5.3.3 Operating Environment

Operating environment is as listed below.:

Operating System — Ubuntu or Microsoft Windows 10

Central Processing Unit (CPU) — Intel Core i5 6th Generation processor or higher. An AMD equivalent processor will also be optimal.

Platform: Anaconda, Tensor-Flow,Jupyter-Notebook, Keras

5.4 External Interface Requirements

5.4.1 User Interfaces

User can give input through a GUI and get the report of the video as a text file which contains details of anomaly if present.

5.4.2 Hardware Interfaces

A desktop with 3.4GHz CPU, 8G memory, good GPU, 32-bit or 64-bit processor, minimum 8-bit graphic adapter and running windows operation system. Basic hardware like monitor, mouse and keyboard are a must for input and output. An additional camera will be required for real-time video processing.

5.4.3 Software Interface

Python with the version 3.6.x with the following modules installed, OpenCV, NumPy, Scikit-learn, SciPy, Keras, TensorFlow, H5py, OS, time, fmatch, pickle. Windows operation system and a Web browser (Chrome preferable). Camera modules installed for importing video file from camera.

5.4.4 Communication Interface

User have to manually give input video to the trained model to check the exact time stamp for anomaly in the given video. In output, model show the time at which anomaly is detected. Or a real time feed is also given through the camera.

5.5 System Features

5.5.1 Functional requirements

The system should process images in the chosen image formats. (jpg, png, bmp, tif) The system shall detect abnormal frames which deviate from normal pattern built using training data and display the exact time from the start at which the abnormal event is detected. The length of the test video can vary and no particular limit is imposed. If the abnormality is not detected in the video, user should be displayed with a message that the video is normal. After the whole video is processed the user should be given an option to see the portion of the video where abnormal event was detected. Only the frames

to be suspected as abnormal should be played as a video. Option to see the abnormal portion of the video any number of times should be given to the user using GUI.

5.5.2 Nonfunctional requirements

Usability:

The system is easy to train and test thus navigates in the most expected way with less delay. Since the algorithm is written in such a way that a lot of parallel computation can be performed, extensively high frame rates can be achieved by inculcating multi-threading. User will be allowed to

add his frames easily and direct testing can be started on those frames. Thus, it is user friendly, reducing complexity on users and getting them a better result in a faster way.

Assumption:

Test frames are either in .tif, .jpg, .png, .bmp formats. Each frame is taken to represent one second of the video. So, a 200 second video is assumed to generate 200 frames. The learning rate for training is

chosen to be 0.0001 and the acceptance value of error in training is set to 0.04. In the testing phase threshold for reconstruction error is assumed to be 0.00000045 for optimal accuracy during testing.

Performance:

Pre-captured video frames will be analyzed fast as a significant amount of time is saved in not capturing the video. On the other hand, real-time capturing will extend the required time.

Reliability:

The software is tested with dataset and the output is very close to the actual abnormal scenarios thus turning out to be reliable when all the assumptions mentions are considered.

5.5.3 Use case description

1. Primary Actor: Crowd

Scenario: Abnormal Event is created by the crowd at any point of time.

2. Primary Actor: CCTV

Secondary Actor: System

Scenario: Crowd Scene is captured by the CCTV and is fed to the system to detect any abnormal event occurring in the video.

3. Primary Actor: System

Scenario 1: Preprocessing - The frames are extracted from the raw videos, resized and processed.

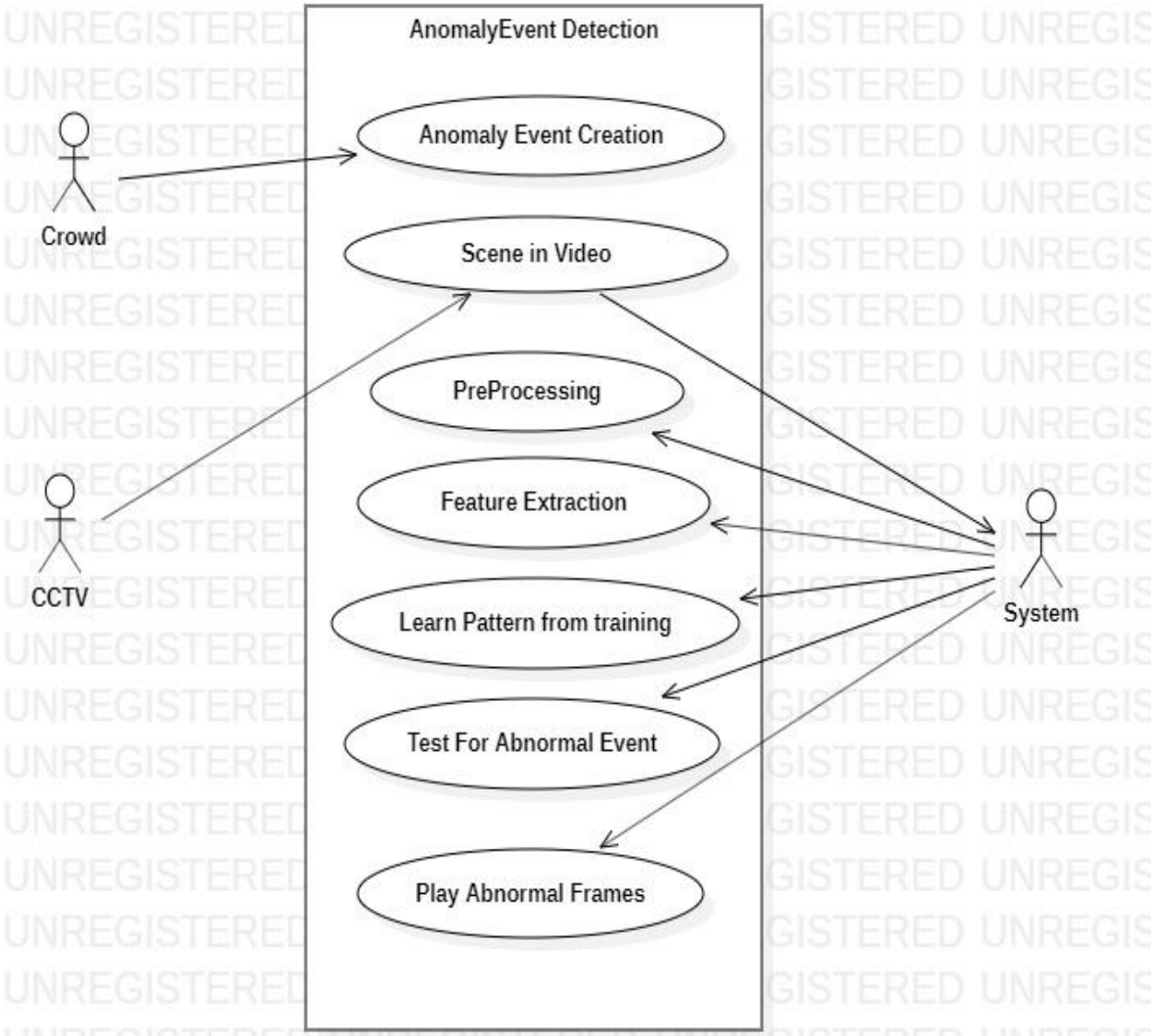
Scenario 2: Feature Extraction – For learning spatial structures of the video frames and patterns from encoded structures.

Scenario 3: Learning Pattern the training set – Learn the normal pattern from the training videos, then anomalies are detected as events deviated from the normal patterns.

Scenario 4: Test for Abnormal Event - The video is tested for the abnormality and the reconstruction error is checked. The time at which the abnormality is detected is also displayed.

Scenario 5: Play Abnormal Frames – Portion of the videos having abnormal event is displayed to the user.

5.5.4 Use case diagram



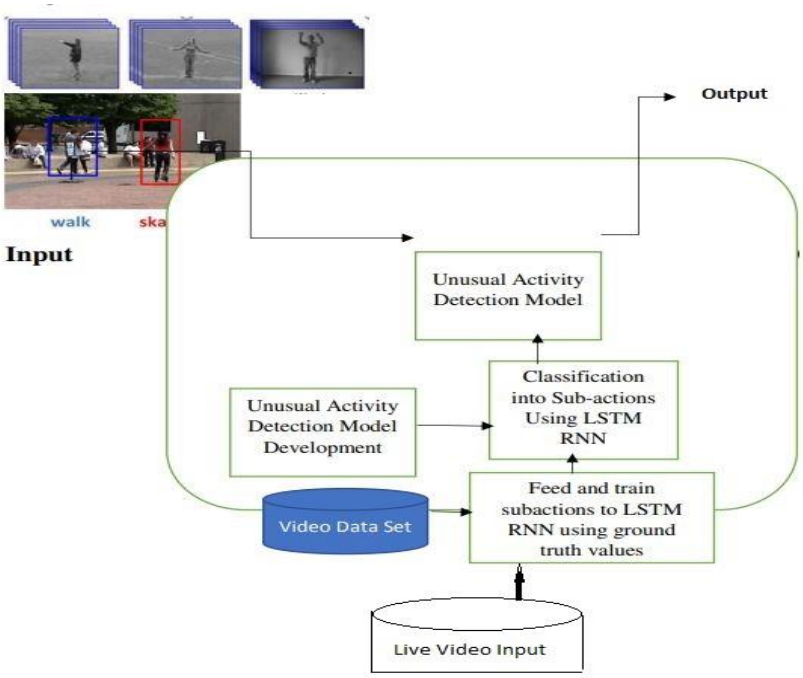
CHAPTER 6

DESIGN

6.1 Introduction

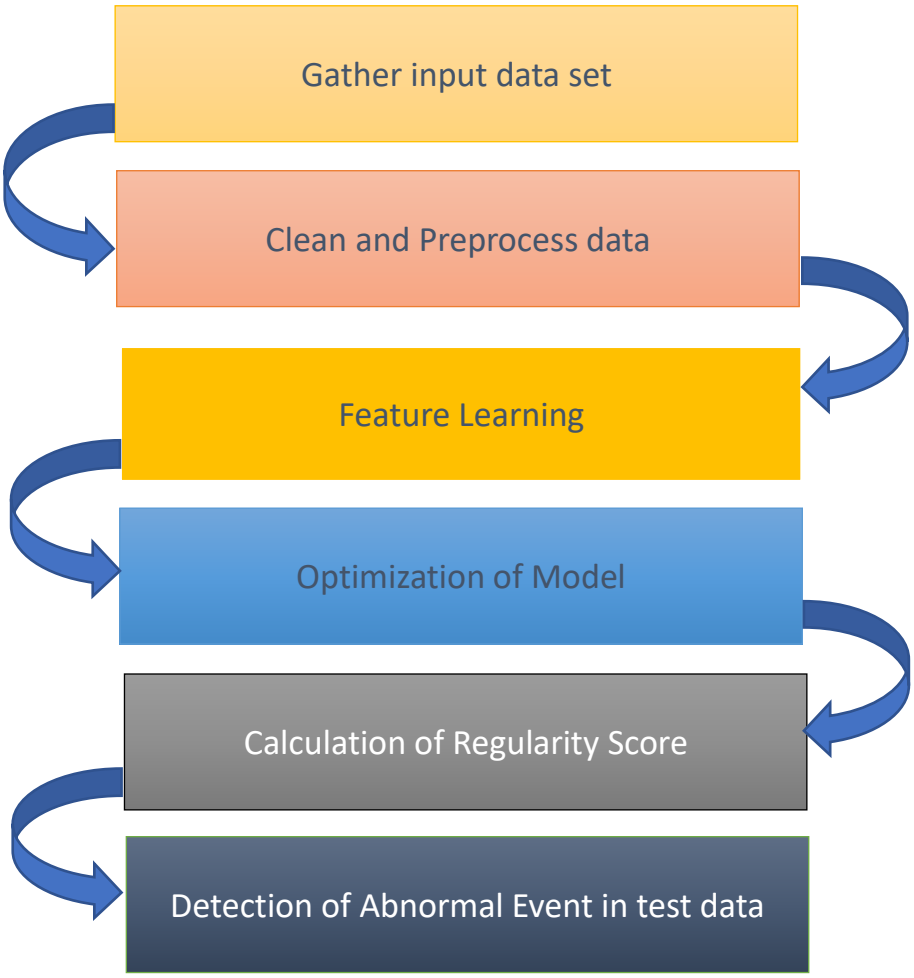
- It is all about the **reconstruction error**.
- We use an autoencoder to learn regularity in video sequences.
- The intuition is that the trained autoencoder will reconstruct regular video sequences with **low error** but will not accurately reconstruct motions in irregular video sequences.

6.2 Architecture Design as per the selected architectural

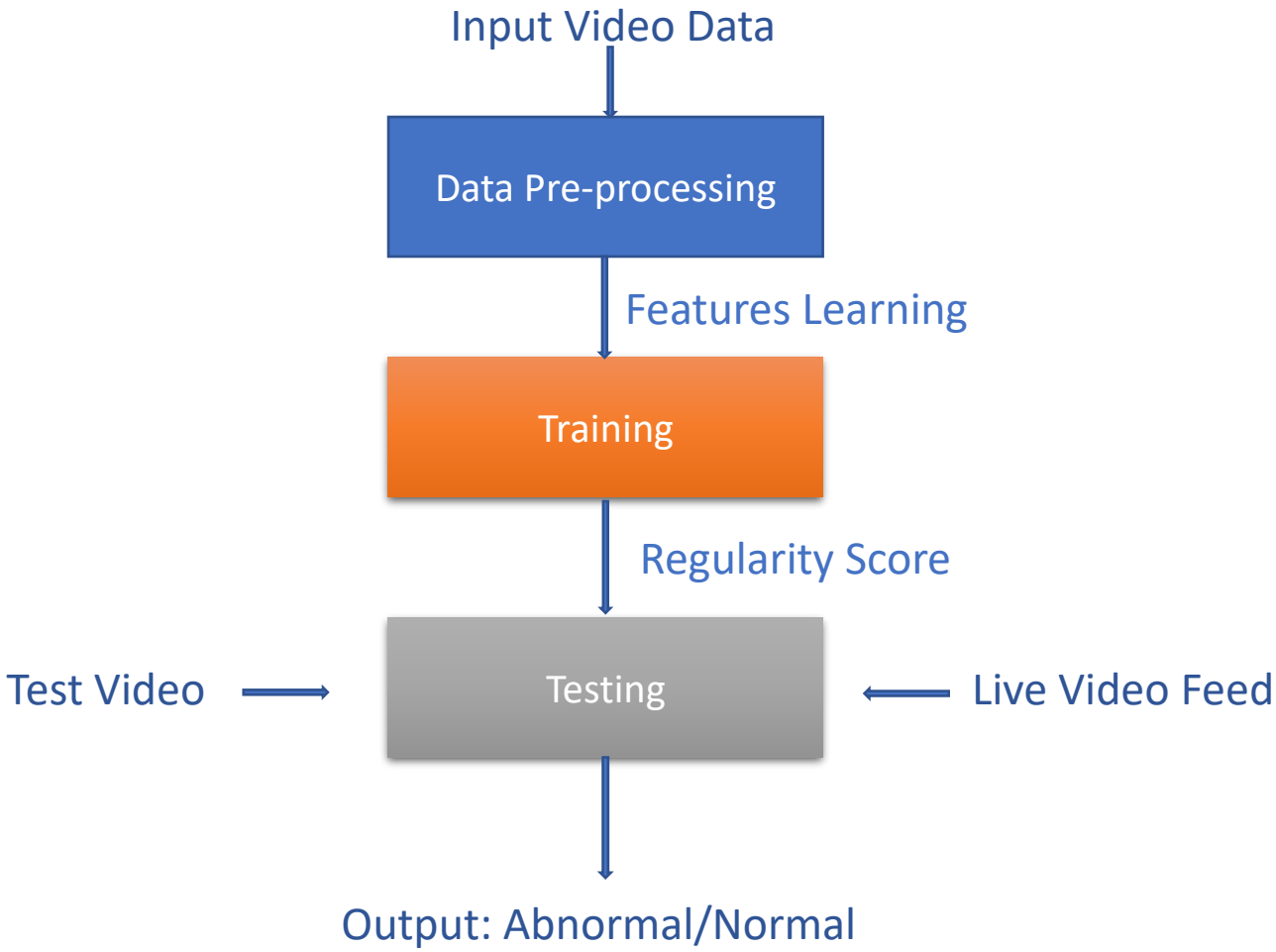


6.3 Low Level Design

Overall Process (Block Diagram)



6.4 Workflow Diagram

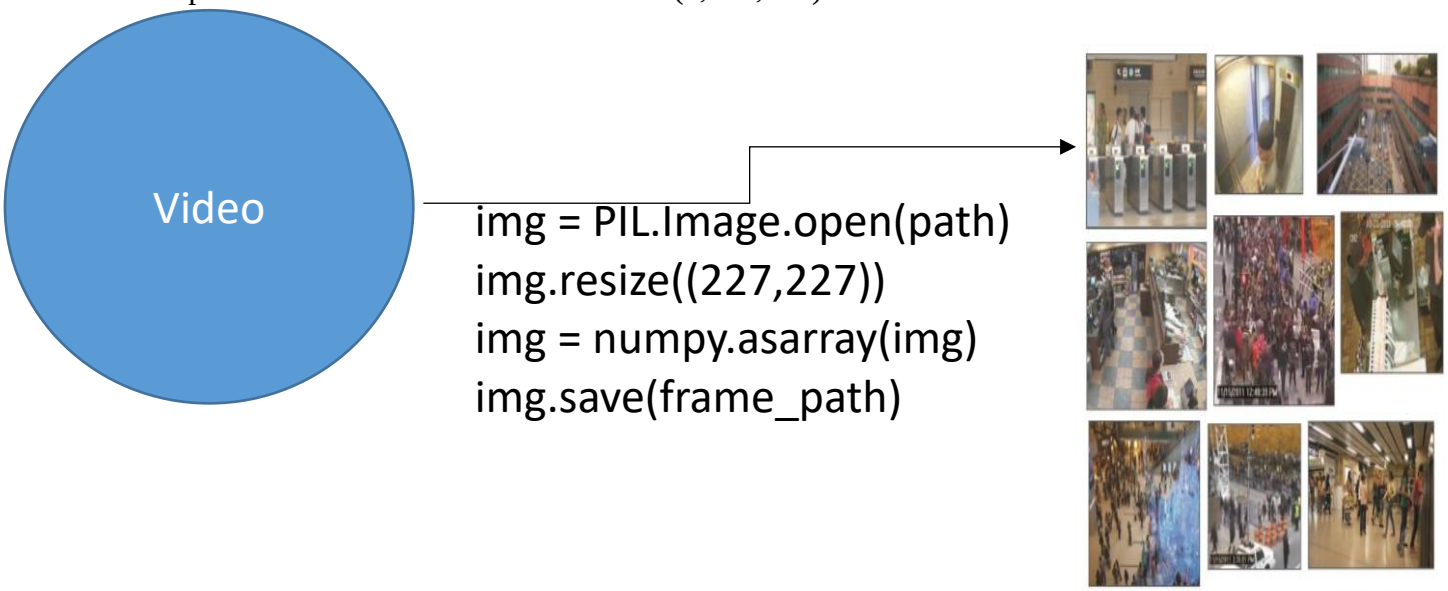


6.5 Data Pre-Processing and Feature Extraction

6.5.1 Data Preprocessing

6.5.1.1 Input: Videos

6.5.1.2 Step 1: convert videos into frame of size (3,227,227)



6.5.1.3 Step 2: For each frame, resize it to 3 scales (227,227, batch_size)

6.5.1.3.1 a,b,c=img.shape()

6.5.1.3.2 img.resize(b,c,a)

6.5.1.3.3 After that, the images are converted to grayscale to reduce dimensionality.

6.5.1.3.3.1 gray=0.2989*img[: , : , 0]+0.5870*img[: , : , 1]+0.1140*img[: , : , 2]

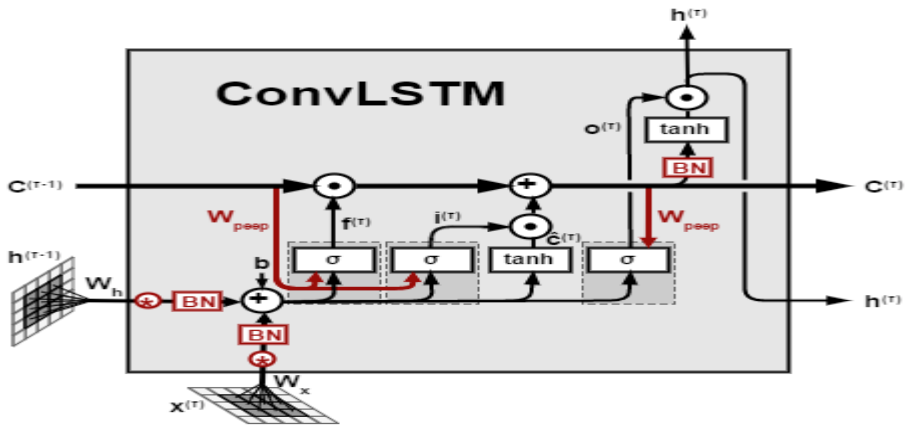
6.5.1.3.4 To ensure that the input images are all on the same scale, the pixel values are scaled between 0 and 1 and subtracted every frame from its global mean image for normalization.

6.5.1.3.4.1 img = np.clip(img,0,1)

6.5.1.3.5 Now thw input is ready for training

6.6 Feature Learning

- We propose a convolutional spatiotemporal autoencoder to learn the regular patterns in the training videos.
- Our proposed architecture consists of two parts|
  - spatial autoencoder for learning spatial structures of each video frame
  - temporal encoder-decoder for learning temporal patterns of the encoded spatial structures.



6.7 Training

6.7.1 Training model Algorithm

```
def get_model(reload_model = True):
    “””
        Parameters
        ----------
        reload_model: bool
            Load Saved model or retrain it
    “””
    if not reload_model:
        return load_model(“path of pretrained model”)
    get the training set created by preprocessing in form of Numpy array
    training_set=np.load('training.npy’)
    load the sequential model from Keras Library
```

seq = Sequential()

add Convolutional LSTM layers to the sequential model with the required input parameters via . add()method.

compile the sequential model using activation function as sigmoid and loss as “mean square error” and optimizer as Adam.

fit the training set into the sequential model of convLSTM filters.

save the model to the defines path.

return the seq model.

6.7.2 Training Explained

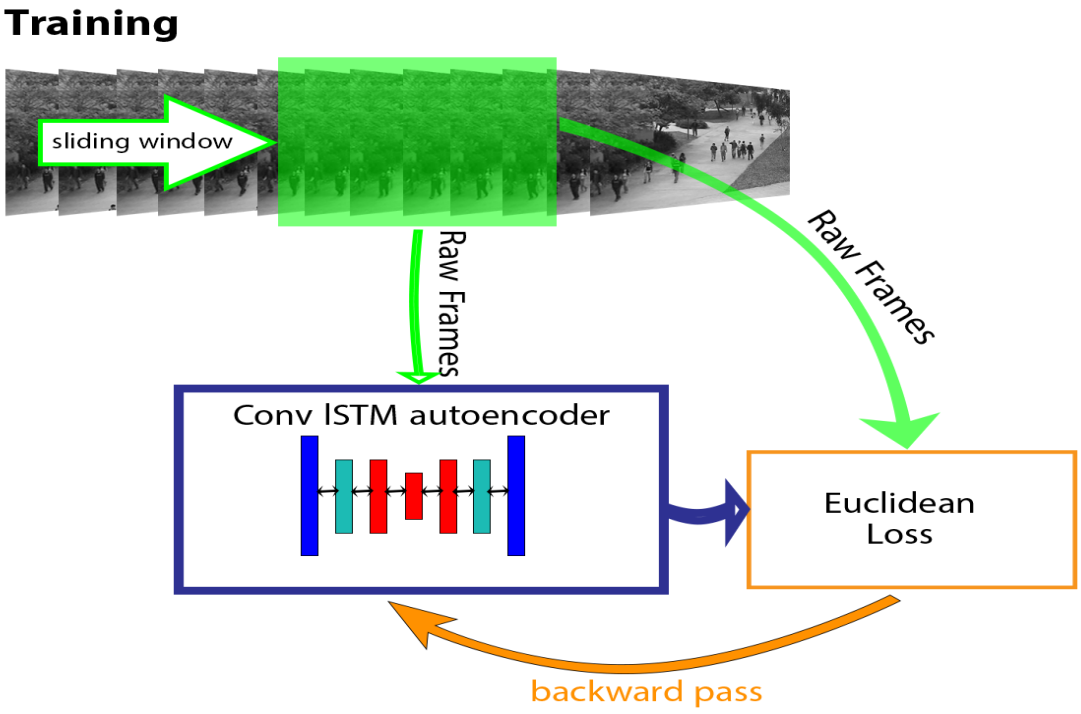
Compared to the usual fully connected LSTM (FC-LSTM), ConvLSTM has its matrix operations replaced with convolutions. By using convolution for both input-to-hidden and hidden-to-hidden connections, ConvLSTM requires fewer weights and yield better spatial feature maps. The formulation of the ConvLSTM unit can be summarized with (1) through (6).

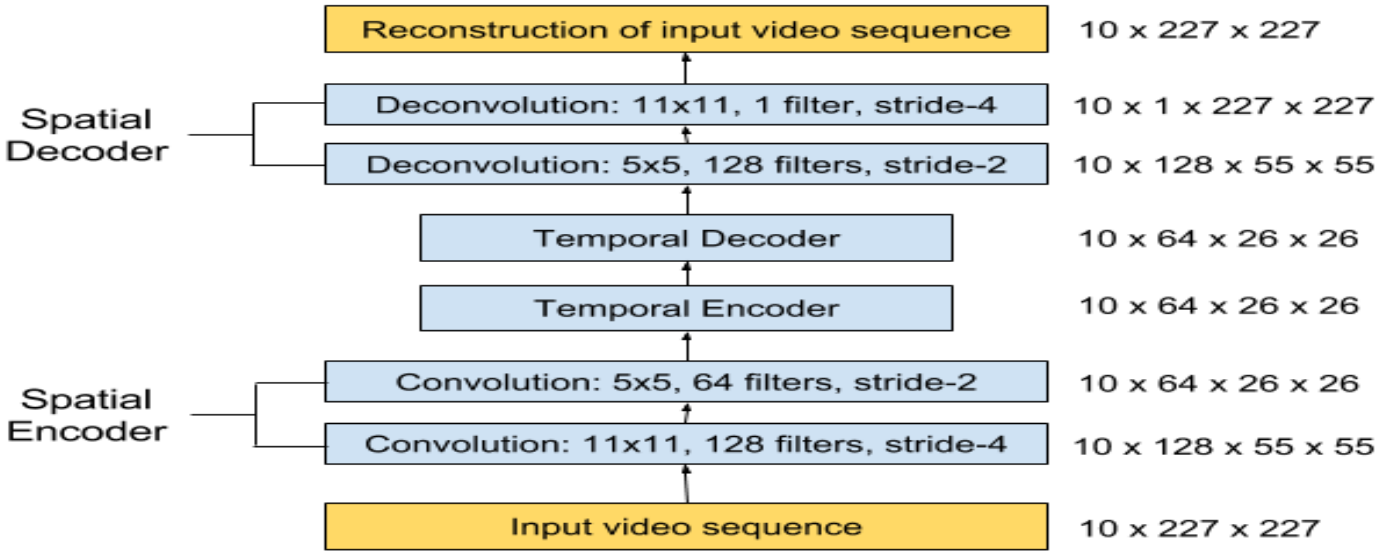
- f(t) = sigmoid(W(f) * [h(t1), x(t), C(t1)] + b(f)) ……...…… (1)
- i(t) = sigmoid(W(i) * [h(t1), x(t), C(t1)] + bi) ……... (2)
- C^(t) = tanh(W(c) * [h(t1), x(t)] +b(c)) ……...…….. (3)
- C(t) = f(t) * C(t-1) + i(t) *C^(t)…….....(4)
- o(t) = sigmoid(W(o) * [h(t-1), x(t), C(t-1)] + b(o)) ……...…… (5)
- h(t) = o(t)*tanh(C(t)) ……..... (6)

- Equation (1) represents the forget layer
- (2) and (3) are where new information is added
- (4) combines old and new information
- (5) and (6) output what has been learned so far to the LSTM unit at the next timestep
- The variable x(t) denotes the input vector
- h(t) denotes the hidden state
- C(t) denotes the cell state at time t

- W are the trainable weight matrices, b are the bias vectors, and the symbol * denotes the Hadamard product.
- The primary purpose of convolution in case of a convolutional network is to extract features from the input image.
- Mathematically, convolution operation performs dot products between the filters and local regions of the input.

- Suppose that we have some n x n square input layer which is followed by the convolutional layer. If we use an m x m filter W, the convolutional layer output will be of size (n-m+1) x(n-m+1).
- A convolutional network learns the values of these filters on its own during the training process, although we still need to specify parameters such as the number of filters, filter size, the number of layers before training.





The spatial encoder takes one frame at a time as input, after which T = 10 frames have been processed, the encoded features of 10 frames are concatenated and fed into temporal encoder for motion encoding. The decoders mirror the encoders to reconstruct the video volume**.**

6.8 Testing

- Once the model is trained, we can evaluate our model's performance by feeding in testing data and check whether it is capable of detecting abnormal events while keeping false alarm rate low.
- The reconstruction error of all pixel values I in frame t of the video sequence is taken as the Euclidean distance between the input frame and the reconstructed frame:

$$e(x,y,t) = \left\| I(x,y,t) - f_W\left(I(x,y,t)\right) \right\|_2$$

$$e(t) = \sum_{(x,y)} e(x,y,t)$$

- Where f_W is the learned weights by the spatiotemporal model. We then compute the abnormality score $s_a(t)$ by scaling between 0 and 1. Subsequently, regularity score $s_r(t)$ can be simply derived by subtracting abnormality score from 1:
    - $s_a(t) = (e(t) - e(t)min)/e(t)max$…… (7)
    - $s_r(t) = 1 - s_a(t)$……...(8)

Testing Algorithm

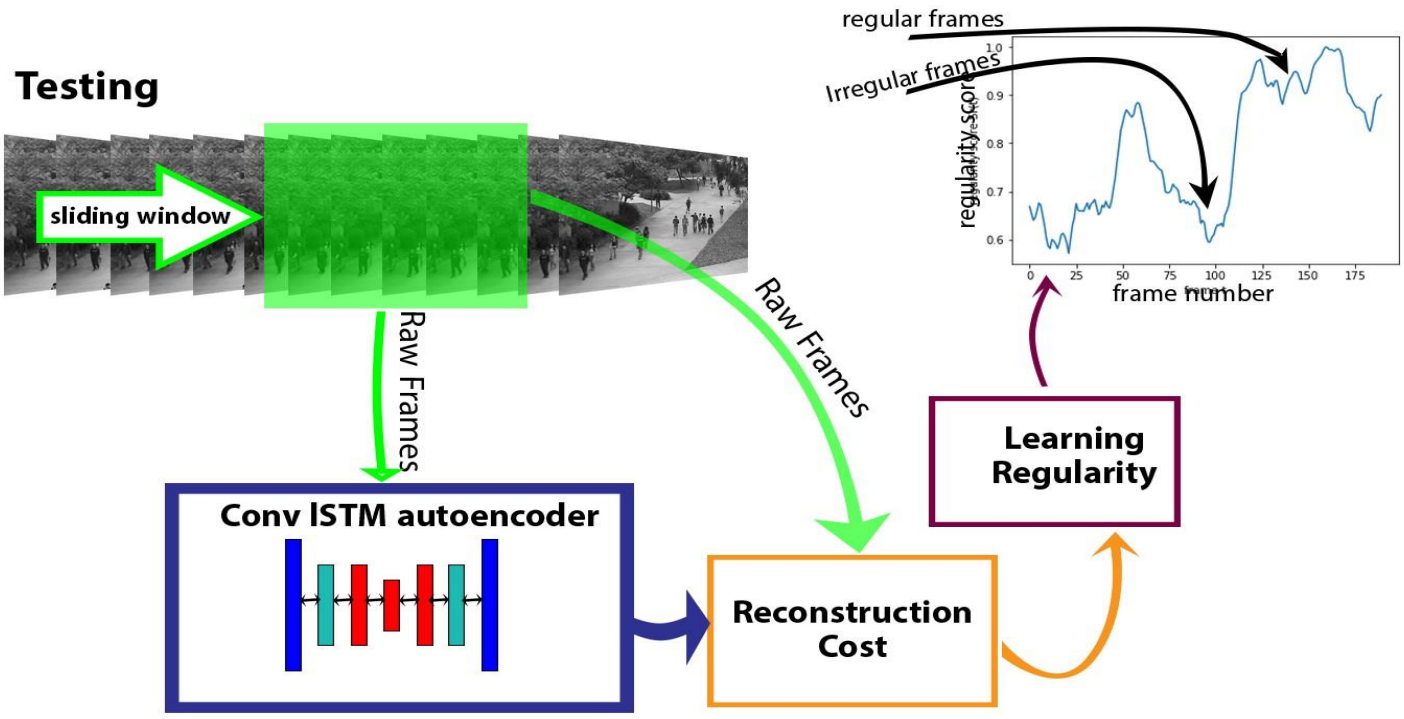
```
def test():
    Get the trained model from saved path
    model = get_model(True)

    Load Test data
    test_data = get_single_test()
    Get the size of frames

    size = test_data.shape()

    Apply the sliding window technique to get the sequences

    Get the reconstruction cost of all the sequences from above equations
    Plot the regularity Score to determine the abnormality in videos
```



6.9 Conclusion

We are formulating this anomaly detection as a spatiotemporal sequence outlier detection problem and to detect this we are using spatial feature extractor and temporal sequencer ConvLSTM to handle the problem. A mean_squared_loss between original frames and predicted bunch of frames is calculated and a threshold is determined above which a frame is predicted to be anomalous.

CHAPTER 7

IMPLEMENTATION

7.1 Tools Introduction and Technology Introduction

Tensor flow:

It is an open source artificial intelligence library, using data flow graphs to build models. It allows developers to create large-scale neural networks with many layers. TensorFlow is mainly used for: Classification, Perception, Understanding, Discovering, Prediction and Creation.

Keras:

Keras is a powerful and easy-to-use free open source Python library for developing and evaluating deep learning models. It wraps the efficient numerical computation libraries Theano and TensorFlow and allows you to define and train neural network models in just a few lines of code.

Matplotlib:

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+.

OpenCV:

OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection. In this tutorial, we explain how you can use OpenCV in your applications.

7.2 Overall view of the project in terms of implementation

Abnormal events are due to either:

1.Non-pedestrian entities in the walkway, like bikers, skaters, and small carts.

2.Unusual pedestrian motion patterns like people walking across a walkway or at the grass surrounding it.

The model that consists of a spatial feature extractor and a temporal encoder-decoder is trained with video volumes consists of only normal scenes, with the objective to minimize the reconstruction error between the input video volume and the output video volume reconstructed by the learned model. It involves the extraction of the required data by resizing each frame into different scales. Each layer is uniformly divided to a set of patches of same size which are non-overlapping. Five continuous frames in of corresponding regions are stacked together to form a spatial-temporal cube. With the cubes, 3D gradient features are computed on each of them. These features in a video sequence are processed separately according to their spatial coordinates. Only features at the same spatial location in the video frames are used together for training and testing. After the model is properly trained, normal video volume is expected to have low reconstruction error, whereas video volume consisting of abnormal scenes is expected to have high reconstruction error. By thresholding, on the error produced by each testing input volumes, our system will be able to detect when an abnormal event occurs.

Preprocessing:

This step includes importing the video frames and making it reading for training. It involves extraction of features which is the input to the training algorithm. Initially each video is converted to frames which approximately denotes the number of seconds.

The frames obtained are then analyzed in steps of 5 frames at a time basis. So 5 consecutive frames are taken at a time and each of the frames are resized to different levels of resolution. From experimental result 3 scales are chosen, one with 20x20 resolution, 30x40 resolution and 120x160 resolution. This form an image pyramid which hold lower resolution images at the top and as we go down the pyramid higher resolution images will be encountered. This process of generating image pyramid by resizing images to different scales helps in analyzing more content at each pixel level. In the top of the pyramid each pixel will hold a bigger portion of the image compared to the lower levels of the pyramid which would hold a zoomed in version of the smaller image covering a smaller portion. This helps the algorithm to detect both local and global abnormal event at each region of the image.

At each scaled image, the image is divided into non-overlapping region of size 10x10 pixels called patch. So based on the resolution different number of patches will be obtained. The whole process is repeated for all the frames.

Corresponding patches in consecutive frames are collected and stacked. So, patches will be stacked over consecutive frames thus forming cubes. The cube thus represents a spatio-temporal value of frames in different parts of the frames. Cubes generated are individually converted to features. Together all of these form the feature set which will be fed to the training algorithm for further processing. We propose a convolutional spatiotemporal autoencoder to learn the regular patterns in the training videos.
The primary purpose of convolution in case of a convolutional network is to extract features from the input image. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data.

7.3 Explanation of Algorithm and how it is been implemented

Training

The features extracted in data pre-processing step is used as an initial input to the training algorithm. We will use Keras to build our convolutional LSTM autoencoder.

To build the autoencoder, we should define the encoder and the decoder. The encoder accepts as input a sequence of frames in chronological order, and it consists of two parts: the spatial encoder and the temporal encoder. The encoded features of the sequence that comes out of the spatial encoder are fed into the temporal encoder for motion encoding.

The decoder mirrors the encoder to reconstruct the video sequence. We use Adam as an optimizer with a learning rate set to 0.0001, we reduce it when training loss stops decreasing by using a decay of 0.00001, and we set the **epsilon** value to 0.000001.

Algorithm:

```
def get_model (reload_model = True):
    “””
        Parameters
        ----------

        reload_model: bool
            Load Saved model or retrain it
    “””

    if not reload_model:
        return load_model(“path of pretrained model”)

    get the training set created by preprocessing in form of Numpy array
    training_set=np.load ('training.npy’)
    load the sequential model from Keras Library
    seq = Sequential ()
    add Convolutional LSTM layers to the sequential model with the required input parameters via seq.add() method
    compile the sequential model using activation function as sigmoid and loss as “mean square error” and optimizer as Adam
    fit the training set into the sequential model of convLSTM filters
    save the model to the defines path
    return the seq model
```

- Compared to the usual fully connected LSTM (FC-LSTM), ConvLSTM has its matrix operations replaced with convolutions. By using convolution for both input-to-hidden and hidden-to-hidden connections, ConvLSTM requires fewer weights and yield better spatial feature maps. The formulation of the ConvLSTM unit can be summarized with (1) through (6).
  - f(t) = sigmoid(W(f) * [h(t1), x(t), C(t1)] + bf)…….(1)
  - I(t) = sigmoid(W(i) * [h(t1), x(t) , C(t1)] + bi)……..(2)
  - C^(t) = tanh(W(c) * [h(t1), x(t)] + b(c))……...(3)
  - C(t) = f(t) * C(t-1) + i(t) * C^(t)…….....(4)
  - o(t) = sigmoid(W(o) * [h(t-1), x(t), C(t-1)] + b(o))…….(5)
  - h(t) =o(t)*tanh(C(t))…….(6)

- Suppose that we have some n x n square input layer which is followed by the convolutional layer. If we use an m x m filter W, the convolutional layer output will be of size (n-m+1)x(n-m+1).

- A convolutional network learns the values on its own during the training process, although we still need to specify parameters such as the number of filters, filter size, the number of layers before training.

7.4 Conclusion

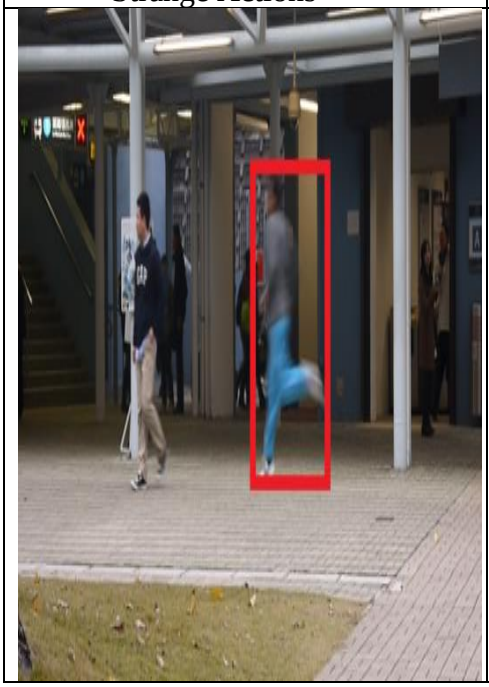
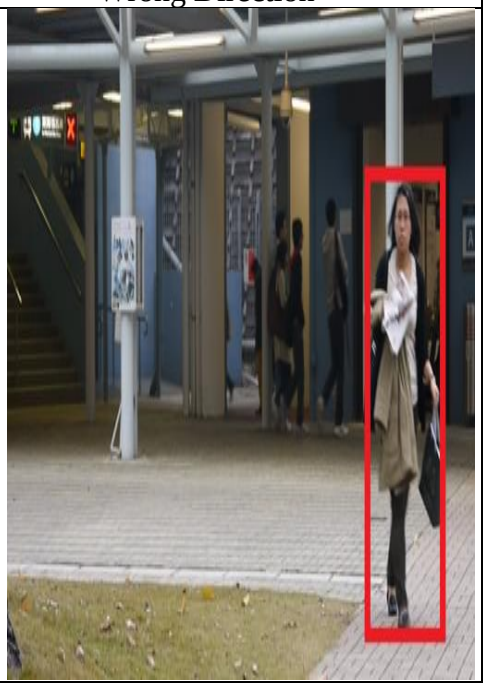
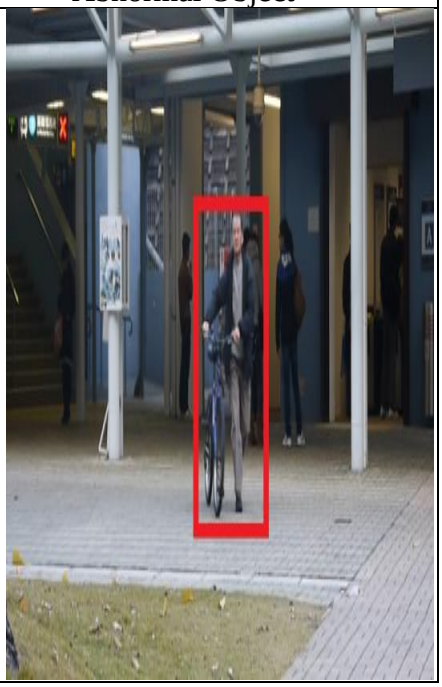
We have used ConvLSTM for implementation of our project to detect the anomalies present in the videos. Our proposed method proves to be efficient than the other traditional methods used for this problem statement. Our implementation has shorter running time with an accuracy 75-80%. Hence it reduces the human effort to detect any abnormalities.

CHAPTER 8

TESTING

8.1 Introduction

Testing of software plays a very important role in evaluation of functionality of a software application with an intent to find whether the developed software met the specified requirements or not. For Anomaly detection in surveillance videos testing includes if our model is robust to any situation or event occurs in the surveillance. There are three kind of anomaly can occur in a video or a place strange actions of people, movement of crowd in wrong direction or an Abnormal object comes in the picture. So, our model must be capable of recognizing and classifying these events.

Strange Actions	Wrong Direction	Abnormal Object
		

8.2 Test Cases

As shown in the following table we tried all the test-cases possible and got satisfactory result except for movement in wrong direction because that depends on the place if that event is abnormal for that situation and place or not.

Test Case Name	Test Procedure	Test Data	Result Expected	Result Found
Strange Action	Feed video of the event in bunches of frames in the form of numpy array	Video containing the Strange actions like running etc.	Return the bunch number of frames containing the abnormal event	Return the bunch number of frames containing the abnormal event
Abnormal Objects	Feed video of the event in bunches of frames in the form of numpy array	Video containing the Abnormal objects	Return the bunch number of frames containing the abnormal event	Return the bunch number of frames containing the abnormal event
Wrong Direction of Movement	Feed video of the event in bunches of frames in the form of numpy array	Video containing the movement of people in wrong direction	Classify event as Abnormal	Classifying event as Normal
Normal Events	Feed video of the event in bunches of frames in the form of numpy array	Video containing the Normal events	Classify the bunches as normal	Classify the bunches as normal

CHAPTER 9

RESULTS & PERFORMANCE ANALYSIS

9.1 Result Snapshots

Fig 9.1 represents our model with 2 layers of Conv3D as Convolution and 2 layers of Conv3d_transpose as Deconvolution with 3 conv_lstm_2d layers for temporal encoding and decoding. This model has over 1m trainable parameters. For training of this model, we need a high RAM and a high processing CPU. We fed this model with 16 training videos after preprocessing and extracting features from them as shown in Fig 9.3 and got a training accuracy of 78 % by training with 100 epochs as shown in Fig 9.4. After training the model testing dataset of same place but with abnormal events is fed and prediction is made on the bunches of frames as shown in Fig 9.5.

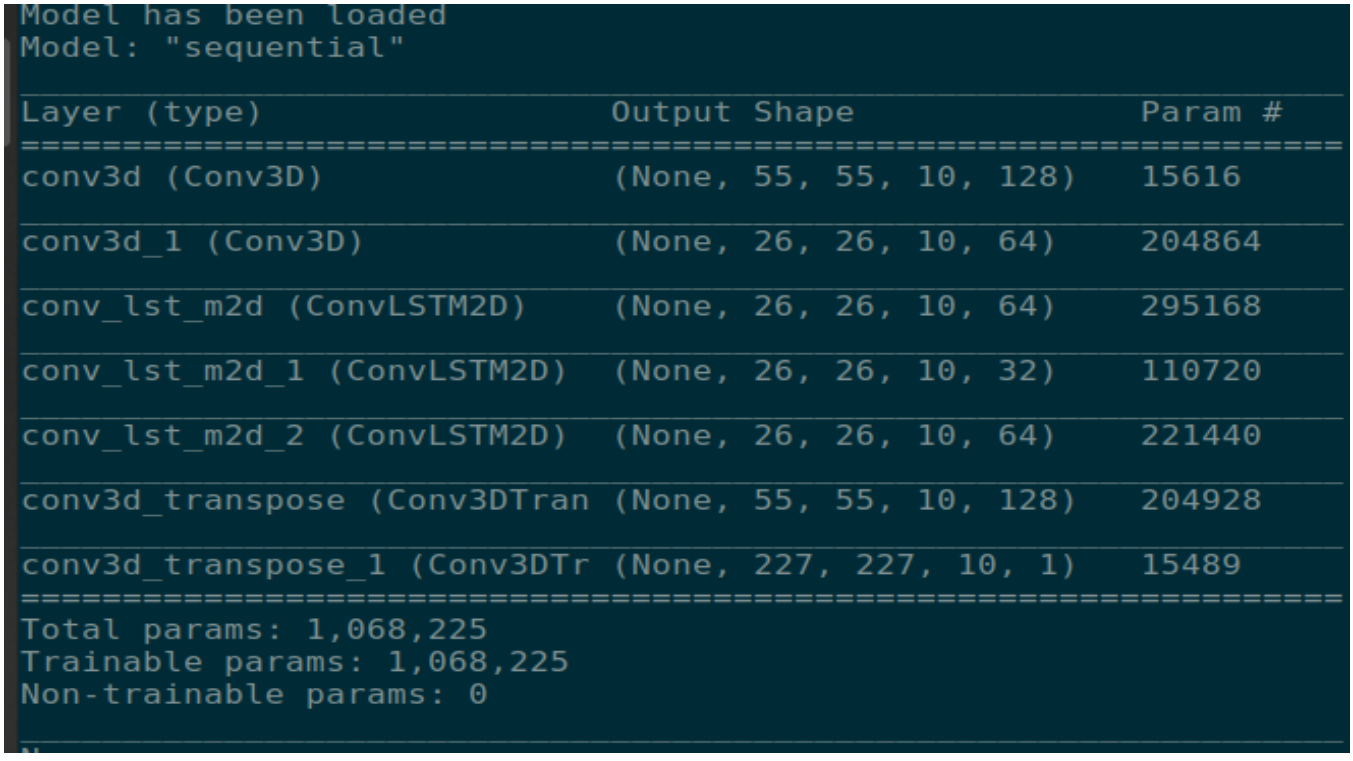
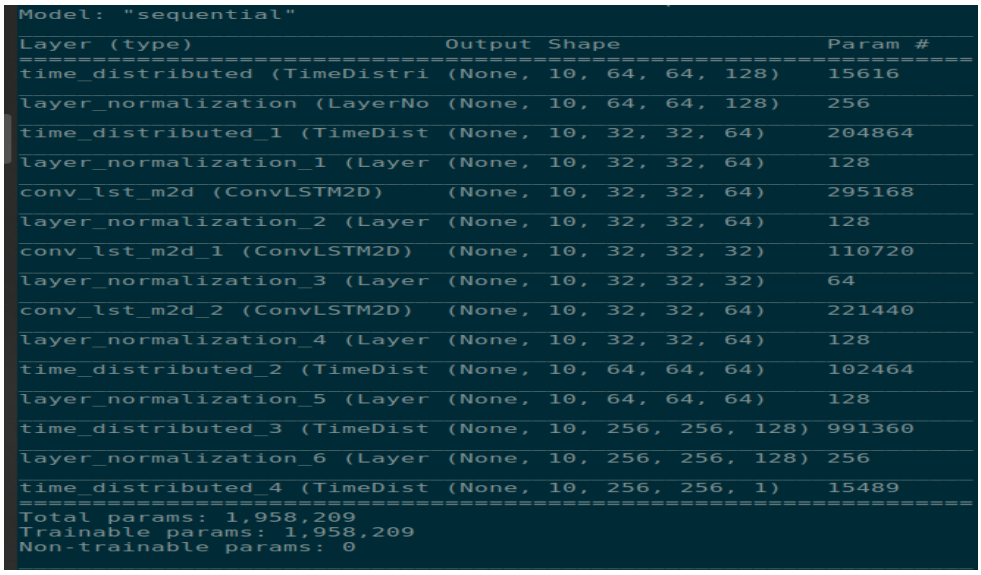


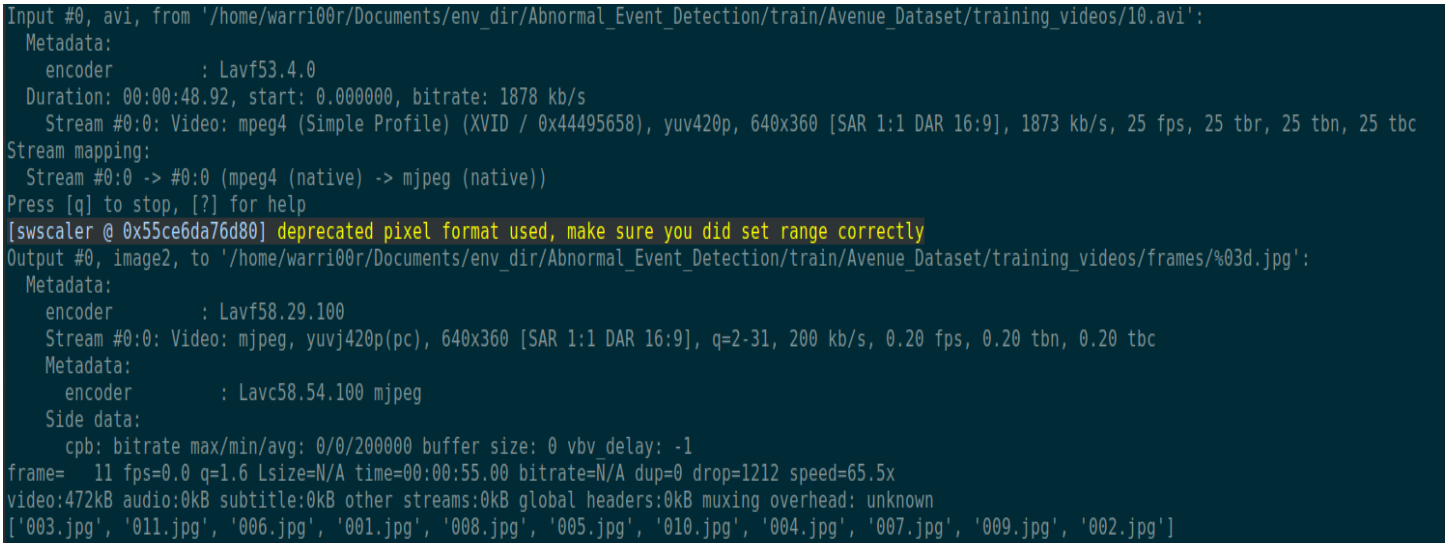
Fig 9.1: Model Summary of Conv3D

Fig 9.2 shows a model with time-distributed Conv2D layers with Layer-Normalization between each temporal encoding and decoding layers. Total trainable parameters of this model are around 2M to train such a large model we need a very high specification and processing CPU and GPU. It also needs a RAM of above 12GB to process this model. We tried this model on our Avenue Dataset and UCSD dataset as Well but didn't get the required results due to lack of such a heavy machine.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
time_distributed (TimeDistri (None, 10, 64, 64, 128)   15616
_________________________________________________________________
layer_normalization (LayerNo (None, 10, 64, 64, 128)   256
_________________________________________________________________
time_distributed_1 (TimeDist (None, 10, 32, 32, 64)    204864
_________________________________________________________________
layer_normalization_1 (Layer (None, 10, 32, 32, 64)    128
_________________________________________________________________
conv_lst_m2d (ConvLSTM2D)    (None, 10, 32, 32, 64)    295168
_________________________________________________________________
layer_normalization_2 (Layer (None, 10, 32, 32, 64)    128
_________________________________________________________________
conv_lst_m2d_1 (ConvLSTM2D)  (None, 10, 32, 32, 32)    110720
_________________________________________________________________
layer_normalization_3 (Layer (None, 10, 32, 32, 32)    64
_________________________________________________________________
conv_lst_m2d_2 (ConvLSTM2D)  (None, 10, 32, 32, 64)    221440
_________________________________________________________________
layer_normalization_4 (Layer (None, 10, 32, 32, 64)    128
_________________________________________________________________
time_distributed_2 (TimeDist (None, 10, 64, 64, 64)    102464
_________________________________________________________________
layer_normalization_5 (Layer (None, 10, 64, 64, 64)    128
_________________________________________________________________
time_distributed_3 (TimeDist (None, 10, 256, 256, 128) 991360
_________________________________________________________________
layer_normalization_6 (Layer (None, 10, 256, 256, 128) 256
_________________________________________________________________
time_distributed_4 (TimeDist (None, 10, 256, 256, 1)   15489
=================================================================
Total params: 1,958,209
Trainable params: 1,958,209
Non-trainable params: 0
```

Fig 9.2: Model Summary of Time Distributed Conv2D



```
Input #0, avi, from '/home/warri00r/Documents/env_dir/Abnormal_Event_Detection/train/Avenue_Dataset/training_videos/10.avi':
  Metadata:
    encoder         : Lavf53.4.0
  Duration: 00:00:48.92, start: 0.000000, bitrate: 1878 kb/s
    Stream #0:0: Video: mpeg4 (Simple Profile) (XVID / 0x44495658), yuv420p, 640x360 [SAR 1:1 DAR 16:9], 1873 kb/s, 25 fps, 25 tbr, 25 tbn, 25 tbc
Stream mapping:
  Stream #0:0 -> #0:0 (mpeg4 (native) -> mjpeg (native))
Press [q] to stop, [?] for help
[swscaler @ 0x55ce6da76d80] deprecated pixel format used, make sure you did set range correctly
Output #0, image2, to '/home/warri00r/Documents/env_dir/Abnormal_Event_Detection/train/Avenue_Dataset/training_videos/frames/%03d.jpg':
  Metadata:
    encoder         : Lavf58.29.100
    Stream #0:0: Video: mjpeg, yuvj420p(pc), 640x360 [SAR 1:1 DAR 16:9], q=2-31, 200 kb/s, 0.20 fps, 0.20 tbn, 0.20 tbc
    Metadata:
      encoder         : Lavc58.54.100 mjpeg
    Side data:
      cpb: bitrate max/min/avg: 0/0/200000 buffer size: 0 vbv_delay: -1
frame=   11 fps=0.0 q=1.6 Lsize=N/A time=00:00:55.00 bitrate=N/A dup=0 drop=1212 speed=65.5x
video:472kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: unknown
['003.jpg', '011.jpg', '006.jpg', '001.jpg', '008.jpg', '005.jpg', '010.jpg', '004.jpg', '007.jpg', '009.jpg', '002.jpg']
```

Fig 9.3: Preprocessing and Feature Extraction

```
Epoch 92/100
184/184 [==============================] - ETA: 0s - loss: 0.0244 - accuracy: 0.7789
Epoch 00092: accuracy improved from 0.77886 to 0.77889, saving model to ./AnomalyDetector.h5
184/184 [==============================] - 122s 661ms/step - loss: 0.0244 - accuracy: 0.7789
Epoch 93/100
184/184 [==============================] - ETA: 0s - loss: 0.0243 - accuracy: 0.7789
Epoch 00093: accuracy improved from 0.77889 to 0.77893, saving model to ./AnomalyDetector.h5
184/184 [==============================] - 122s 665ms/step - loss: 0.0243 - accuracy: 0.7789
Epoch 94/100
184/184 [==============================] - ETA: 0s - loss: 0.0243 - accuracy: 0.7790
Epoch 00094: accuracy improved from 0.77893 to 0.77898, saving model to ./AnomalyDetector.h5
184/184 [==============================] - 126s 683ms/step - loss: 0.0243 - accuracy: 0.7790
Epoch 95/100
184/184 [==============================] - ETA: 0s - loss: 0.0242 - accuracy: 0.7790
Epoch 00095: accuracy improved from 0.77898 to 0.77904, saving model to ./AnomalyDetector.h5
184/184 [==============================] - 122s 665ms/step - loss: 0.0242 - accuracy: 0.7790
Epoch 96/100
184/184 [==============================] - ETA: 0s - loss: 0.0242 - accuracy: 0.7791
Epoch 00096: accuracy improved from 0.77904 to 0.77911, saving model to ./AnomalyDetector.h5
184/184 [==============================] - 125s 680ms/step - loss: 0.0242 - accuracy: 0.7791
Epoch 97/100
184/184 [==============================] - ETA: 0s - loss: 0.0242 - accuracy: 0.7791
Epoch 00097: accuracy improved from 0.77911 to 0.77913, saving model to ./AnomalyDetector.h5
184/184 [==============================] - 125s 682ms/step - loss: 0.0242 - accuracy: 0.7791
Epoch 98/100
184/184 [==============================] - ETA: 0s - loss: 0.0241 - accuracy: 0.7792
Epoch 00098: accuracy improved from 0.77913 to 0.77919, saving model to ./AnomalyDetector.h5
184/184 [==============================] - 125s 682ms/step - loss: 0.0241 - accuracy: 0.7792
Epoch 99/100
184/184 [==============================] - ETA: 0s - loss: 0.0241 - accuracy: 0.7792
Epoch 00099: accuracy improved from 0.77919 to 0.77924, saving model to ./AnomalyDetector.h5
184/184 [==============================] - 126s 683ms/step - loss: 0.0241 - accuracy: 0.7792
Epoch 100/100
184/184 [==============================] - ETA: 0s - loss: 0.0240 - accuracy: 0.7793
Epoch 00100: accuracy improved from 0.77924 to 0.77931, saving model to ./AnomalyDetector.h5
184/184 [==============================] - 121s 658ms/step - loss: 0.0240 - accuracy: 0.7793
```

Fig 9.4: Training and Model Accuracy

```
loss:0.0003999914155046875
Bunch Normal
loss:0.0004061610289760421
Anomalous bunch of frames at bunch number 1
loss:0.0003976881120847178
Bunch Normal
loss:0.0004003552702978037
Anomalous bunch of frames at bunch number 3
loss:0.0004070481960274909
Anomalous bunch of frames at bunch number 4
loss:0.0003980714428188688
Bunch Normal
loss:0.0003978530719395989
Bunch Normal
loss:0.000393362636687385
Bunch Normal
loss:0.0003816379070284530
Bunch Normal
loss:0.0003843763896666728
Bunch Normal
loss:0.0003854880728543643
Bunch Normal
loss:0.0003870097956355677
Bunch Normal
loss:0.0003820153324840795
Bunch Normal
loss:0.0003877336345748113
Bunch Normal
loss:0.0003816906401924558
Bunch Normal
loss:0.0004016438026118354
Anomalous bunch of frames at bunch number 15
loss:0.0003943344760242780
Bunch Normal
loss:0.0003955563417099789
Bunch Normal
loss:0.0003953349693846231
Bunch Normal
```

Fig 9.5: Testing Results and Predictions

CONCLUSION

In this research, we have successfully applied deep learning to the challenging video anomaly detection problem. We formulate anomaly detection as a spatiotemporal sequence outlier detection problem and applied a combination of spatial feature extractor along with temporal sequencer ConvLSTM to tackle the problem. The ConvLSTM layer not only preserves the benefits of FC-LSTM but is additionally suitable for spatiotemporal data thanks to its inherent convolutional structure. By application of convolutional feature extractor in both spatial and temporal space into the encoding-decoding structure, we build an end-to-end trainable model for detecting the video anomaly. The advantage of our model is that it's semi-supervised – the sole ingredient required may be a long video segment containing only normal events in a fixed view. Despite the model's ability to detect abnormal events and its robustness to noise, counting on the activity complexity within the scene, it's going to produce more false alarms compared to other methods.

In future work we would like to train our model on different datasets with different optimizers to check and compare performance and robustness on noise. We will investigate how to improve the result of video anomaly detection by active learning - having human feedback to update the learned model for better detection and reduced false alarms. One idea is to add a supervised module to the current system, with the supervised module works only on the video segments filtered by our proposed method, then train a discriminative model to classify anomalies when enough video data has been acquired.

References

[1]:Kelathodi Kumaran Santhosh, Debi Prosad Dogra, ParthaPratim Ro "Anomaly Detection in Road Traffic Using Visual Surveillance: A Survey" in arXiv:1901.08292v1 [cs.CV] 24 Jan 2019

[2]: Oluwatoyin P. Popoola ,Kejun Wang "Video-Based Abnormal Human Behavior Recognition"—A Review Article in IEEE Transactions on Systems Man and Cybernetics Part C (Applications and Reviews) · November 2012

[3]: Sagar S. Mane, Prof. HemangiSatam, and Prof. V. B. Gaikwad, "Abnormal event detection in video using appearance and motion information," International Research Journal of Advanced Engineering and Science, Volume 3, Issue 3, pp. 223-225, 2018.

[4]: Raksha S, B G Prasad "Anomalous Human Activity Recognition in Surveillance Videos"
International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-2S7, July 2019

[5] Wang, Lin & Zhou, Fuqiang & li, zx & Zuo, Wangxia & Tan, Haishu. (2018). Abnormal Event Detection in Videos Using Hybrid Spatio-Temporal Autoencoder. 2276-2280. 10.1109/ICIP.2018.8451070.

[6] Chong, Yong Shean & Tay, Yong Haur. (2017). Abnormal Event Detection in Videos Using Spatiotemporal Autoencoder. 189-196. 10.1007/978-3-319-59081-3_23.

[7] E. Duman and O. A. Erdem, "Anomaly Detection in Videos Using Optical Flow and Convolutional Autoencoder," in *IEEE Access*, vol. 7, pp. 183914-183923, 2019. doi: 10.1109/ACCESS.2019.2960654

[8] Zhou, Fuqiang & Wang, Lin & li, zx & Zuo, Wangxia & Tan, Haishu. (2019). Unsupervised Learning Approach for Abnormal Event Detection in Surveillance Video by Hybrid Autoencoder. Neural Processing Letters. 10.1007/s11063-019-10113-w.

[9]Bansod, Suprit&Nandedkar, Abhijeet. (2019). Crowd anomaly detection and localization using histogram of magnitude and momentum. The Visual Computer. 10.1007/s00371-019-01647-0.

[10]M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni and N. Sebe, "Abnormal event detection in videos using generative adversarial nets," Beijing, 2017, pp. 1577-1581.

[11]Cong, Yang & Yuan, Junsong& Liu, Ji. (2013). Abnormal event detection in crowded scenes using sparse representation. Pattern Recognition. 46. 1851–1864. 10.1016/j.patcog.2012.11.021.

[12] Waqas Sultani1 , Chen Chen2 , Mubarak Shah2 1Department of Computer Science, “Real-world Anomaly Detection in Surveillance Videos”, University of Central Florida (UCF) arXiv:1801.04264v3 [cs.CV] 14 Feb 2019

[13] “Abnormal event detection in crowded scenes based on deep learning” Zhijun Fang & Fengchang Fei & Yuming Fang & Changhoon Lee & Naixue Xiong & Lei Shu & Sheng Chen(2018). DOI 10.1007/s11042-016-3316-3

[14] Ming Xu , Xiaosheng Yu , Dongyue Chen, Chengdong Wu and Yang Jiang 2”An Efficient Anomaly Detection System for Crowded Scenes Using Variational Autoencoders” 14 August 2019

[15]” An On-Line and Adaptive Method for Detecting Abnormal Events in Videos Using Spatio-Temporal ConvNet” Samir Bouindour , Hichem Snoussi , Mohamad Mazen Hittawe and Nacef Tazi and Tian Wang 21 February 2019