DBS

Dublin Business School

excellence through learning

ABNORMAL EVENT DETECTION FROM THE VIDEO USING RECURRENT NEURAL NETWORKS (RNN)

Dissertation submitted in part fulfillment of the requirements for the degree of

“MSc in Data Analytics” at Dublin Business School

Lakshmi Harika Chennamsetty

10515217

DECLARATION

I, **“Lakshmi Harika Chennamsetty”,** declare that this research is my original work and that it has never been presented to any institution or university for the award of Degree or Diploma. In addition, I have referenced correctly all literature and sources used in this work and this work is fully compliant with the Dublin Business School’s academic honesty policy.

ACKNOWLEDGMENTS

I owe my sincere gratitude and thanks to my professor and guide **Dr. Sharam Azizi** for guiding me throughout the process of implementing and documenting.

I also would like to acknowledge my friends and seniors support who guided me with the knowledge they possess.

I would like to extend my warm gratitude towards my family and well-wishers who have been a great support morally and made me achieve and reach this stage.

ABSTRACT

Setting up the surveillance camera system has become inevitable in all the places throughout the world. Huge number of human resources has been deployed for monitoring the surveillance system in which the unusual happens very rarely. Also, It requires manual operation to review the footage for any abnormal incidence. Taking this as a use case, this thesis work designated to provide the initial solution for the same with the machine learning techniques to avoid involving human resource for monitoring any abnormal activities which got identified in the broadcast video from the surveillance system.

To identify the abnormal activities, it requires to relate the video frames in the past and the current to predict for the unprecedented activities in the video frame. Considering the nature of the required machine learning model, Long Short-Term Memory (LSTM) has been chosen which is of a type recurrent neural networks (RNN). The basic feature of the chosen algorithm is to persist data for the future use and make any decisions by considering the information stored in the memory. Handling the temporal data is one of the main reasons behind choosing the LSTM algorithm. This paper aims at developing the solution, only to identify the explosions in the video and later shall be enhanced.

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 1: INTRODUCTION

Video surveillance is one of the major areas that people started using throughout the world for security reasons and monitoring the activities. It helps in monitoring the activities of both the commercial and residential places so that we shall avoid criminal activities. On the other hand, a wide range of research is in progress in performing the automation of identifying the abnormal activities and use cases like marking attendance with identifying the face, etc. Machine learning has evolved to a greater extent in the field of identifying the objects, identifying events in the video and use libraries like OpenCV to extract the essential information from the video frames.

1.1 BACKGROUND

Video surveillance camera set-up in public places has become a mandatory one nowadays. Say, a Surveillance camera is almost installed in places like commercial sites, residential flats & apartments, public places like roads, theatres, shopping malls, etc. Just recording the videos and saving the same for future reference doesn't alone gives the actual benefits of setting up the video surveillance system. We need to have an advanced machine learning models which are capable enough to take an action with respect to the events broadcasting from the live video streaming. With respect to different places, the need for installing and monitoring the video from the surveillance camera differs.

For a scenario of setting up the video surveillance camera in the park is different from setting up the same in the classroom. In the classroom, we shall use the surveillance camera for even marking attendance by identifying the faces recognized inside the classroom. Whereas in the case of the park, it's purely for security reasons. From the discussed scenarios, we shall say that Machine learning features with the combination of video surveillance system going to perform an evolution by feeding the knowledge of identifying the activities from the broadcast video and helps us to perform the countermeasure activities. If machine learning is not in the picture, we need to involve a human being for surveillance of the videos broadcasted from the camera. The background of this thesis is to perform the analysis of the abnormal activities like identifying the explosions very instantly from the video captured.

1.2 ANOMALY DETECTION – MOTIVATION AND CHALLENGES

A common need of analyzing the real-world activities like anomalies from the video frames becomes inevitable due to its importance of taking the action measures in the event of unexpected happenings like bomb blasts and other explosions. Many cases have been identified in the world where any anomaly incident was not intimated to the external world immediately after the incident due to the heavy damage caused by the explosion. Considering this has a motivation, the objective of this thesis work is automating the intimation of any such anomalies happened in a particular place with the help of a video surveillance camera, which doesn't require any involvement of human beings to perform the intimation. This system then shall be propagated to other areas of identification like earthquakes, theft, etc.

The major challenge faced over identifying the anomalies is making use of the right machine learning algorithm for performing such detection over the input video frames. On the other hand, training the model with the video samples of the explosion using unsupervised machine learning is considered a challenging part of the implementation. The unsupervised learning technique for identifying any kind of anomaly detection has proved to be the best approach considering the input data, performance, etc. and to be more specific about the use cases it has been chosen as a mechanism for fraud detections. We have used deep learning mechanisms in the unsupervised learning architecture using SpatioTemporal auto-encoder for performing the implementation of anomalies detection.

1.3 HOW SPATIOTEMPORAL AUTO-ENCODER WORK?

Autoencoders were used for dimensionality reduction. The autoencoder results in the reduced error count when compared to the other principal components used for data clustering. Also, the autoencoder results in better separation of the classification. In performing the analysis on the computer vision and images, Autoencoders have taken a major place in terms of building the different range of applications keeping Autoencoders has the base. Autoencoders are also used for performing the transformation of the images into different versions by performing the modifications in the color scheme. On top of that, autoencoders are used to remove the noise on the images. The tools used to detect outliers have autoencoders that have the base and it uses linear algebra mathematical formula to perform the transformations.

The model is more efficient has it has a greater number of layers rather than having one thick layer to perform the processing. The autoencoder shows better performance when the model works with more complex data and non-linear.

1.4 RESEARCH QUESTIONS

- How deep learning technology helps to analyze and identify anomalies in the videos broadcasted from the surveillance camera?
- How anomaly event detection using machine learning model will be helpful for the real-world use cases?
- Why Do We Apply Dimensionality Reduction - autoencoder to Find anomalies in the videos?

1.5 OBJECTIVE

The main objective of building the model is to develop an Artificial Neural Network model to perform the identification of abnormal events like explosions from the video broadcasted in the image. The base idea is to process the videos by splitting into image frames and apply SpatioTemporal machine learning using autoencoder to find the abnormal changes in the frames which we have captured from the surveillance system. This work takes the videos captured during the bomb blast as the training input data in the form of videos and builds the model with the given input. Due to limitations in the environmental set-up and development time, the scope is limited to handle only with Explosion detection, and this idea later shall be propagated to other abnormal event detections like an earthquake, driving the vehicle in the pedestrian path, disobeying the traffic rules, thief trying to snatch the belongings of others, etc.

The next chapter which is a Literature review helps us to understand the concepts used behind identifying the abnormal activities by analyzing the changes in the video frames broadcasted. Also, the literature review states the aim of the research paper, background information on the machine learning techniques used and other related use cases.

This research work will definitely be the best solution to attain the possibility of creating a solution that will perform countermeasure activities for any abnormal events happened in a particular area without any human intervention on the same. The further phases of this research paper will expand to identify different other abnormal activities and publish the same in the real-time environment as a working solution.

CHAPTER 02: LITERATURE REVIEW

A literature review is a document that helps to narrate the understanding of the topic which we have chosen for the thesis as a topic of study. It is very important that we study all the related areas of the chosen topic and we should be in the situation to answer questions related to the study and experts are believing with the literature review a person shall accomplish it. The literature review also takes you to an in-depth evaluation of the specific research topic. Any study requires understanding the related work so that we shall identify the mistakes in the existing system and think about the solution to the problem solution in a different way which will address the issues identified in the related work.

2.1 LITERATURE PAPERS

(Vu et al., 2017) In this paper the author has researched on Energy-based Localized Anomaly Detection in Video Surveillance as it is one of the important tasks in the research applications. Here, the author has used unified framework on Restricted Boltzmann Machine known as RBM one of the booming unsupervised learning techniques. The author focuses on unsupervised learning techniques for this abnormal detection in videos due to growing data that cannot be defined as what is abnormal. This proposed methodology directly focuses on the image pixels and learns new representation just by learning without the need of labels. The model reconstructs the detected abnormal locations based on the reconstruction errors. The author has built this model in such a way that it supports both offline and online streaming. Offline method, it just detects the abnormal locations within the streams, whereas online detection is that even when the video is updated incrementally with new video data being arrived. The author makes another three models to setup a benchmark for performance and accuracy comparison. The author proves that this proposed model detects the abnormality in the video just by using the image pixels thus outperforming other state of the art approaches.

(Amer et al., 2013), The author has proposed his research based on the topic “Enhanced One Class Support Vector Machines for Unsupervised Anomaly detection”. In this thesis, the author focuses on the effectiveness of unsupervised learning for outlier detection. One class Support Vector Machine is one of the machine learning algorithms that is extended from the original support vector machine algorithms concept which is ultimately used for unsupervised learning techniques. Unsupervised learning techniques work directly on any unseen data knowing the fact that there are sparsely outliers present in it. The author uses two methods by modifying the state-of-the-art approach. One is robust one class SVM and the other one eta one class SVM. He also used two other techniques from semi supervised learning methods namely – sub space division and Gaussian Kernel. Sub spaced division is used for high dimensional detection of anomalies and Gaussian Kernel is to improve the effectiveness of the Support Vector Model that highly relies on the Kernel properties. The author here compares his model with other nine of the state-of-the-art approaches. Here, out of the four models created, eta one class support vector machine outperforms all the other models. Sub spaced approach did not show much of significant improvement.

(Xu et al., 2015) In this research paper the author proposes “Learning deep representations of appearance and Motion for anomalous detection”. The unique things that the author has used here is that 1) choosing a complex video dataset 2) instead of using hand crafted appearance and motions as features, the author uses DeepNEt technique that utilizes deep neural network learning that automatically learn the features. 3) The author uses a two-way fusional approach that benefits both early fusion and late fusions. 4)Stacked denoising auto encoders to learn both appearance and representations as well as early fusion data. Frame level and Pixel level detection of the anomalies are captured, and the performance metrics used here are ROC, AUC and Equal error Rate. One class Support vector machine learning technique is used to predict the anomalies detected scores and integrates it called as late fusion binding strategy for the final anomaly detection. The author uses two commonly available datasets to explore his ideology. The two datasets have been experimented with the proposed techniques and has shown better results among all other existing methods and techniques. Apart from this the author concludes that the early fusion technique shows less performance results compared to late fusion technique.

(Hasan et al., 2016), In this paper the author implements an understanding of temporal regularity in a video sequence. The author implements this problem by first understanding regular motion patterns by creating a very generic model. Later, he proposes two methods that are built and works with no supervision on autoencoders. In the first phase the author uses spatio temporal local features and then tries to learn the auto encoder on them. In the second phase of the implementation, the author encompasses a fully convolutional feed forward auto encoder, that learns the frame end to end. The author makes sure that model works for multiple datasets. The author learns temporal regularity by extracting handcrafted appearance and motion features. These features are sent as input to the convolutional layers via auto encoders. Low-level Motion Information in a small temporal cuboid will be processed. Trajectory encoding is another technique implemented that processes local features. These local features are processed by encoding with Histograms of Optical Flows and Histograms of Oriented Gradients forming the final motion features. The author visualizes the regularities in frames, pixels. The author compares the quantitative analysis made for auto encoders and trajectories and proves that this technique outperforms the state-of-the-art techniques.

(Luo et al., 2017), The author in this paper – “A revisit of Sparse coding based Anomaly detection in Stacked Recurrent neural network framework” inspired by the sparse coding based anomaly detection tries to implement a Temporal coherent Sparse coding technique. Here, the author encodes similar neighboring frames with similar reconstruction coefficients. Temporal coherent sparse coding is mapped with stacked Recurrent neural network. The author states that this research paper works manifold - 1) Encoding temporal coherent sparse coding with Stacked RNN the performance of the model is accelerated and 2) The dataset used is largest of all the existing datasets and states the complexity that this model works with. The author uses metrics like run time and the performance of the model to compare it with other existing models in the technical state. When run time is considered, the model takes about an hour to train the data and takes about 0.02 seconds to detect an anomaly in a single frame. The prediction of anomalies using only temporal coherent sparse coding is ten times slower than stacked RNN. With this author proves that Stacked RNN outperforms all other state of the art models present.

(Pradhan et al., n.d.), The author in this paper researches about Anomaly Detection using Artificial Neural Network and tries to explain the tools, techniques and evaluation metrics followed to qualify the model's performance. User behavior is chosen as a parameter in anomaly intrusion detection. The author uses back propagation neural network technique to experiment this. There are two things the author focuses on the data – 1) To check if the model is capable of classifying normal traffic as expected 2) To detect known and unknown attacks irrespective of the data size. The author uses these metrics to quantify the model. True positive rate – When anomalies are detected as it is, False positive rate – Classifies normal scene as an anomaly, True negative – detects the normal scene as normal, False negative – Classifies anomaly as normal and finally Accuracy which is overall calculation of the previously mentioned metrics. Accuracy is defined as the number of correct classifications divided by over all instances. According to metrics there was a 0% of false positive rate which means none of the normal frames were classified as attacked but however, one attack went undetected. The author has shown 88% accuracy for the proposed model and believes that this model is better than the existing models.

(Devaraju and Ramakrishnan, n.d.), In this paper – "Detection of Accuracy for Intrusion Detection System using neural networks", the authors aims to deal with the problems of network by detecting the intrusions using different classifier techniques. The authors have proposed five different classifiers namely – 1) Feed

Forward Neural Network (FFNN) 2) Elman Neural Network (ENN) 3) Generalized Regression Neural Network (GRNN) 4) Probabilistic Neural Network (PNN) 5) Radial Based Neural Network (RBNN). The authors use accuracy as a metrics to qualify the model's performance. The data is reduced using feature reduction techniques and the author proves that model with reduced data performs better than the model that process with full featured data set. The authors have proposed principal component analysis as one of the data detection methods to reduce the data such that important features of the data are captured and sent to the models as an input. The following are the performance score for the different classifiers used. – 1) Feed Forward Neural Network shows 79.49% 2) Elman Neural Network shows 78.1% 3) Generalized Regression Neural Network shows 58.74% 4) Probabilistic Neural Network shows 85.56% 5) Radial Based Neural Network shows 83.51%.

(Aneetha and Bose, 2012), In this paper – " The combined techniques for anomaly detection using neural networks and clustering techniques" the authors have aimed to achieve a fast, robust, accurate and also an effective model to detect anomalies in the communication world. The proposed algorithm is Self-Organized mapping algorithm (SOM) that starts with null network and gradually grows with the data. In simple SOM weights are declared randomly. In this paper, the author initializes the initial vectors, updating the neighborhood rules and learning rate dynamically as the data grows. The clustering algorithms K-Means is used for grouping the similar nodes of Modified SOM into number of K clusters that uses similarity measures. The parameters used are 1) Distances threshold 2) Connection strength and 3) Neighborhood functions. The performance of the models is measured using accuracy. The modified SOM has improved the performance of the detection by 2%. When clustering is involved the performance is again decreased by 1.5%, however updating the neighborhood function has been improved by connection strength and when the learning rate increase the number of output node decreases. The overall detection of attacks rate is shown as 98.5% with all the above-mentioned steps implemented. With this the author states that the overall false detection rate is only 2% and this model is one of the most reliable models to detect the anomalies.

(Ho, n.d.), In this paper the authors aim to explain the advancements in the deep neural network for detecting anomalies, object detection, image recognition and some of the other practical experiments that are being carried out using neural networks. The neural networks significantly increase the throughput, performance and the accuracy of the models. The author compares the performance of the deep learning techniques for various experiments. The author also states that anomaly detection can be best done with unsupervised hierarchical feature extraction. When neural networks are combined with feature deduction techniques the performance is even more better compared when the models are based only on the neural networks. The authors are specific about implementing dimensional reduction as it yields better performance. Here the authors use, Independent component analysis with non-parametric Bayesian approach. This method determines the most optimal features that can be used for latent feature vector thus avoiding the heavy labor in parameter tuning that most of the deep learning techniques entail. ICA with neural networks have substantially increased the performance 10% more compared to all the other models.

(Malhotra et al., 2016), In this paper the author – “ LSTM based Encoder Decoder for Multi-Sensor anomaly detection ”is trying to implement Long Short-Term Memory. The author feels that there are external factors that goes uncaptured by the sensors which are highly unpredictable. This, the author feels as one of the greatest challenges in detecting the anomalies using standard approaches. Here, the author proposes a new technique – Long Short-Term memory networks based Encoder and Decoder (EncDe – AD) to predict normal time series behavior. The author shows that the proposed methodology is one of robust as it can detect anomalies in any given situation like – predictable, unpredictable, periodic, aperiodic and even in quasi periodic time series. Using LSTM, the author says that the anomalies can be detected in short term and long term data as well. The author uses different datasets of different data sizes to show that the model works best and is robust for any given data size and for any type of time series. The author uses True

Positive and False positive rate for all the datasets to quantify the performance metrics of the model that is being applied on various type of datasets.

(Chong and Tay, n.d.), In this paper – "Modelling Representation of Videos for Anomaly Detection using Deep learning – A Review", the author tries to explain how existing models help in understanding in detecting anomalies in a video sequence. The author says that even a weakly supervised learning model can help in detecting the anomalies better compared to extracting the right features manually. Here, the author reviews all the existing models that are available in deep learning techniques to provide a better understanding of anomalies detection and action recognition. The author explains about the feature extraction models like 1) Conditional RBM and Space Time Deep Belief Network 2) Independent Component Analysis and its variants 3) Deeply Learned Slow Feature Analysis and Gated models and compares the performance of these models using accuracy as performance evaluation metrics. Feature extraction and tracking is much needed when there are dense trajectories. ICA and RICA show very promising results compared to other state of the art models like HOG and HOG Mining in terms of extracting distinct features even from raw videos.

(Agrawal and Agrawal, 2015), In this paper – "Survey on Determining Anomaly Detection using Data mining Techniques", the author states that with lot of techniques available for detecting anomalies in videos there are still loopholes that are existing. Here, to determine the various attacks the author have proposed clustering data mining techniques like 1) K-Means 2) K-Medoids 3) EM Clustering 4) Outlier Detection Algorithms and Classification based anomaly Detection techniques like – 1) Classification Tree, 2) Genetic Algorithm, 3) Neural Networks, 4)Support Vector Machines. The author also explains a few hybrid techniques like Cascading supervised techniques and combined supervised and unsupervised techniques. The author says that although deep learning techniques provide better results, hybrid approaches are giving promising results in terms of accuracy, performance and run time.

(Hasan et al., 2016), In this paper the author implements an understanding of temporal regularity in a video sequence. The author implements this problem by first understanding regular motion patterns by creating a very generic model. Later, he proposes two methods that are built and works with no supervision on autoencoders. In the first phase the author uses spatio temporal local features and then tries to learn the auto encoder on them. In the second phase of the implementation, the author encompasses a fully convolutional feed forward auto encoder, that learns the frame end to end. The author makes sure that model works for multiple datasets. The author learns temporal regularity by extracting handcrafted appearance and motion features. These features are sent as input to the convolutional layers via auto encoders. Low-level Motion Information in a small temporal cuboid will be processed. Trajectory encoding is another technique implemented that processes local features. These local features are processed by encoding with Histograms of Optical Flows and Histograms of Oriented Gradients forming the final motion features. The author visualizes the regularities in frames, pixels. The author compares the quantitative analysis made for auto encoders and trajectories and proves that this technique outperforms the state-of-the-art techniques.

(Luo et al., 2017), The author in this paper – “A revisit of Sparse coding-based Anomaly detection in Stacked Recurrent neural network framework” inspired by the sparse coding-based anomaly detection tries to implement a Temporal coherent Sparse coding technique. Here, the author encodes similar neighboring frames with similar reconstruction coefficients. Temporal coherent sparse coding is mapped with stacked Recurrent neural network. The author states that this research paper works manifold - 1) Encoding temporal coherent sparse coding with Stacked RNN the performance of the model is accelerated and 2) The dataset used is largest of all the existing datasets and states the complexity that this model works with. The author uses metrics like run time and the performance of the model to compare it with other existing models in the technical state. When run time is considered, the model takes about an hour to train the data and takes about 0.02 seconds to detect an anomaly in a single frame. The prediction of anomalies using only temporal coherent sparse coding is ten times slower than stacked RNN. With this author proves that Stacked RNN outperforms all other state of the art models present.

2.2 CONCLUSION FROM THE LITERATURE REVIEW

To conclude, in the literature review, we have analyzed and documented all the outline existing knowledge related to the topic "Abnormal Event Detection". By analyzing the related works and other related subjects, we have made the decision to go with Recurrent Neural Networks (RNN) for the implementation purpose and to be more specific use Long Short-Term Memory (LSTM) for the autoencoder model creation. For materializing the chosen algorithm, we have decided to go with Keras libraries with the neural networks as a platform for building the proposed machine learning model. Technical details are mentioned in detail in the implementation chapter of the document.

With reference to all the above studied papers, analysis and understood we are not going to perform the same model but just extract the idea from these papers. From all the papers, it is clearly understood that hybrid or combinational approaches provide better results compared to standalone approaches. Deep learning networks provide much better results. In this research paper of detecting anomalies in the video sequence we are to implement Recurrent neural networks along with Long

Short-Term memory technique that will capture pixel data even when there is large video dataset so that not even a single frame is missed during the process.

CHAPTER 03: METHODOLOGY

This chapter gives a more detailed description on the methodology followed to develop the machine learning model starting from data preprocessing, methodologies involved in preparing the model to perform the analyzing and with the evaluation methodologies used for evaluating the model. On the other hand, we have also explained the methodology used in terms of project execution starting from requirement understanding, which is understanding the problem case of our thesis, identifying the problems in implementing the algorithm and ending up in materializing the same.

3.1 DATA PREPROCESSING

As a first step, we have chosen the proper dataset to train the model with the video clips. Please find the dataset link given below. The chosen dataset has video clips classified into different types like Abuse, Arrest, Arson, Assault, Burglary, Explosion and Fighting, which all listed has abnormal activities. Due to limitations in the system configuration and lack of time for the development, we have chosen only explosion event detector for the thesis. On the other end, the machine learning model is also trained videos with normal activities which will result in negative when we try to identify the abnormality in the video clip input.

Dataset:

https://www.dropbox.com/sh/75v5ehq4cdg5g5g/AABvnJSwZI7zXb8_myBA0CLHa?dl=0

Data preprocessing is an initial step which involves a series of steps like reading a video, extracting frames, handle the same in the python and perform the calculation on the screen time required for processing the video. To perform the data preprocessing, we import the video, read the content, extract the frames from the video input and save them as images in the temporary memory for processing the same. Post to the frame's extraction, apply machine learning model which we have developed to perform the predictions on identifying any abnormal events in the image frames. Repeat the data preprocessing step for all the data frames. We divide our dataset into training dataset and testing dataset, in which we will be training model using the training data and check for the performance of the model in terms of identifying the model with the test data. On the other hand, the possibility of having multiple anomaly incidence in a single clip is also possible and our machine learning model should be capable enough to identify abnormal events even under the mentioned case.

3.2 METHODOLOGY

This part of the document explains the methodology involved in developing the model. We have followed CRISP-DM methodology to perform the analysis and materialize the analyzed use case into a machine learning algorithm and build a web application by consuming the machine learning model so that it shall be used in the real world. The implementation methodology started with building the machine learning model, proceed with determining the cut points and fetch the summary statistics by clustering the video frames to identify the anomaly detection.

CRISP-DM methodology is a structural based approach for implementing any machine learning or data mining projects. Following the stated methodology, helps us to develop the application in an organized way. The phases of the CRISP-DM methodology process are stated here. The list of phases is Business understanding, Data understanding, Data preparation, Modeling, Evaluation and Deployment. As a process of CRISP-DM methodology, setting the objective of the project by having the desired output is one of the primary steps which we need to describe before we start with the development process. Once we set the object, we shall move towards setting up the project plan which gives a better idea of the execution project plan, tool used for the development and techniques involved. On the other hand, it's very important to think about the solution from the business point of view and decide the business conditions which should be minimally satisfied to consider the taken project has a successful rather then ending up in failure.

The first stage of the CRISP-DP is to understand the business perspective which means discovering the objectives and constraints of the project which we have chosen for our thesis. It’s very important that we are not ignoring this step as it may end up building a totally wrong machine learning solution. Proceeding to stage two, which is data understanding phase, we understand the input data which we are going to use for training the model and identifying any special tool required to prepare our dataset. In our case, understanding the video inputs is essential to differentiate the explosions and bomb blasts recorded in the video clips which we have chosen for modelling. Proceeding to stage three, which is data preparation phase, We prepare the data to feed the same for our machine learning model, which is an abnormal event detector in our case. For processing, its video clips inputs in our case and we shall prepare the video clip data by removing unwanted segments from the video which is nowhere going to help our model to get trained for prediction.

The Keras high-level neural networks API has been used to develop the model with deep machine learning mechanism for this thesis. The Keras library provides the flexibility of using the API in a faster way and with modular approach. Also, it supports convolutional networks for processing the images/video clips, which we have used in our implementation to identify the abnormal event in the videos. On top of that, our application works with both CPU and GPU.

Anomaly identification from the Video surveillance camera is one of the most complex and challenging problems. On top of that, backgrounds, motions and objects in the video frames need to be cluttered. The ideal solution to perform such analysis uses the method of local spatial regions to perform the identification of anomalies. We use an autoencoder deep learning neural network model to identify the explosions in the video clips by making the videos into image frames. The machine learning model uses dense neural network cells which is one of the types in the autoencoder models. And the autoencoder model is built with Long Short-Term Memory (LSTM). long short-term memory is one of the types of recurrent neural networks (RNN) and its basic feature is to persist data for future use and make any decisions by considering the information stored in the memory. Handling the temporal data is one of the main reasons behind choosing the LSTM algorithm.

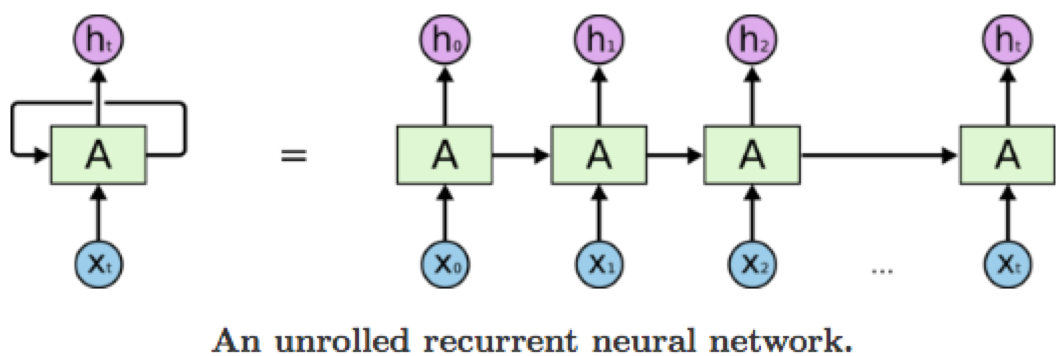

Figure 1: An unrolled recurrent neural network

The autoencoding algorithm is used to perform compression and decompression. The autoencoder need to train to perform the compression on the data which it got trained for. The compression type should be a specific type, which means if the algorithm is trained to perform compression on video frames of the explosions, the defined autoencoder would work only if the selected feature is explosion and it won't work with other types. Providing appropriate training data is the only required item to train the autoencoder and it's required to perform any additional coding or engineering. The autoencoder can be built with having a list of the items like an encoding function which performs the actual compression, a decoding function which performs the decompression of the data and a distance function which calculates the distance between the volume of the information loss and the data represented in the compressed format and the decompressed representation of data. We have different mechanisms to minimize the reconstruction loss measured at the distance function. On the other hand, the autoencoders is really very simple to understand and make use of it in practical.

The built application uses Spatio-Temporal AutoEncoder, which is based on deep neural networks. The main idea of using autoencoder is to learn the representation in the video frames and extracting the required features from the video frames. On top of that, it extracts information from both the temporal dimensions and spatial dimensions to perform the 3-D convolutions.

We have used LSTM (long short-term memory) which is a recurrent neural network (RNN) architecture used to develop models in deep learning. The reason behind choosing LSTM (long short-term memory) is, the model observes the differences between the frames and identify if any drastic changes in the processed video frame when compared to the previous items. Usually, LSTM network takes 3 inputs to perform the analysis, the first parameter X is the input of the current time step, the second parameter H is the output from the previous LSTM unit and the third parameter is the C which is the memory of the previous unit and its one of the important data that is considering in the model execution. To summarize, the LSTM always takes input from the previous units and process the same to make the decision accordingly. The other areas where LSTM networks are proven to be the best fit to form the solutions are speech recognition, translation of the text and sequential processing of data to identify any abnormal activities identified in the sequences. The convolutional LSTM network supported in Keras API has been used to develop a model for identifying the abnormal activities in the video frames. In our demonstration, we have restricted to use analyzing the model to extract the feature of bomb blasts/explosions and performed many experiments on both the videos with no abnormal events and videos with explosions and bomb blasts.

CHAPTER 04: IMPLEMENTATION AND RESULT

In the implementation chapter, detailed procedures that we have followed has been mentioned. With respect to the CRISP-DM methodology,which we are following for developing machine learning model, the implementation phase starts once we are done with data understanding and data preparation phase. The terminology Modeling denoted the actual implementation of the model. With reference to the business requirement and the dataset which we have, the right libraries and algorithms have been already decided for the implementation and in this part of the document, we have mentioned the detailed information in materializing the same.

4.1 INTEGRATED DEVELOPMENT ENVIRONMENT (IDE)

We have developed the application using PyCharm which is an integrated development environment and it is specifically used for developing the Python application. The integrated development environment which we have chosen provides easy option to develop the code in Python, adding new files, execute the model in the terminal and setting up the virtual environment for executing the Python application. On top of that, the IDE provides enhanced option to add a Python flask web application for the same.

4.2 WEB APPLICATION TO ACCESS MODEL

To enable the users to interact with our machine learning model to identify any abnormal event has been identified in the video clip, we have created a web application using python flask web framework. The developed application basically has code like HTML page, which has the design for the user interface in which the users can upload the video in the left-hand panel and possible to view the output in the right-hand panel. By this way, even without any coding knowledge the user shall interact and test the application. Next to that, the project has machine learning model, which is the core component. Also, we have cascading style sheet and JQuery to have enhanced design in the user page and to enable the interaction with the application by any button click using scripts. The web application, interacts with the model using the application programming interface methods has a communication channel which takes the user given input to the backend, perform the model execution and return the result to the user interface. In our case, the video clip is the user input and returning the GIF image path which shows any major changes or abnormal activities in the input data.

4.3 LIBRARIES USED

We have used libraries like Keras, Tensorflow, Numpy, OpenCV,Scipy, Matplotlib and Flask for the implementation.

Keras is an exculsive python library and a package that supports neural network techniques. It has provision to execute TensorFlow and other software libraries having Keras as the base. It contains a greater number of commonly used functionalities of the neural-network and formed a tool to process image and text data. The Keras high-level neural networks API has been used to develop the model with deep machine learning mechanism for this thesis. The Keras library provides the flexibility of using the API in a faster way and with modular approach. Also, it supports convolutional networks for processing the images/video clips, which we have used in our implementation to identify the abnormal event in the videos. On top of that, our application works with both CPU and GPU.

TensorFlow was developed by Google for internal Google use. It is used for performing the research and developed at Google for its own products. This is one of the best libraries in performing the symbolic math calculation. TensorFlow is mainly used for developing machine learning applications such as neural networks.

NumPy is a library used for python programming language. It is mainly used to perform enhanced mathematical calculation on huge data, multi-dimensional arrays, multi-dimensional matrices. NumPy is open-source software.

OpenCV (Open Source Computer Vision) is a library mainly used to perform operations in real-time computer vision. It mainly works on processing the image, analyzing the video clips or video broadcasted from the camera and identifying the face and object detection from the video or image. OpenCV also supports the TensorFlow which is a deep learning mechanism.

Scipy is a library which uses NumPy to perform the mathematical calculations. It is used to perform different calculations in scientific programming, linear algebra and other equation solving. SciPy make use of NumPy library to perform calculations on the arrays.

Matplotlib is a Python programming language used to perform plotting. It uses NumPy library for mathematical calculations. It provides application programming interface for creating and embedding the plots in an object-oriented manner.

Python Flask (Web Framework) is used to develop web application interface in Python app. It doesn't depend on any tools or libraries. The flask microframework is used to develop simple web application with HTML user interface to provide the user to interact with the application.

4.4 IMPLEMENTATION IN DETAIL

The visual features have been extracted from the FC layer, which means fully connected layer on the Convolutional 3D network. The input images or video frames are resized to 240 x 320 pixels and set the processing rate to 30 frame rate frequency, which decides the frames appear on a display. Proceeding to the computational part, Convolutional 3D network for every sixteen frames from the clip of video with 12 normalization. To extract the features for the model from the video clips, we take the average of 16-frame clip within the given segment. We take the extracted features from the video clips and pass on to the 3-layer fully connected neural network. The 1st fully connected layer has 512 units followed by 32 units and one unit fully connected layers. we used dropout regularization between fully connected layers. At this point of development of the model, we have not ended up in attaining the maximum level of accuracy.

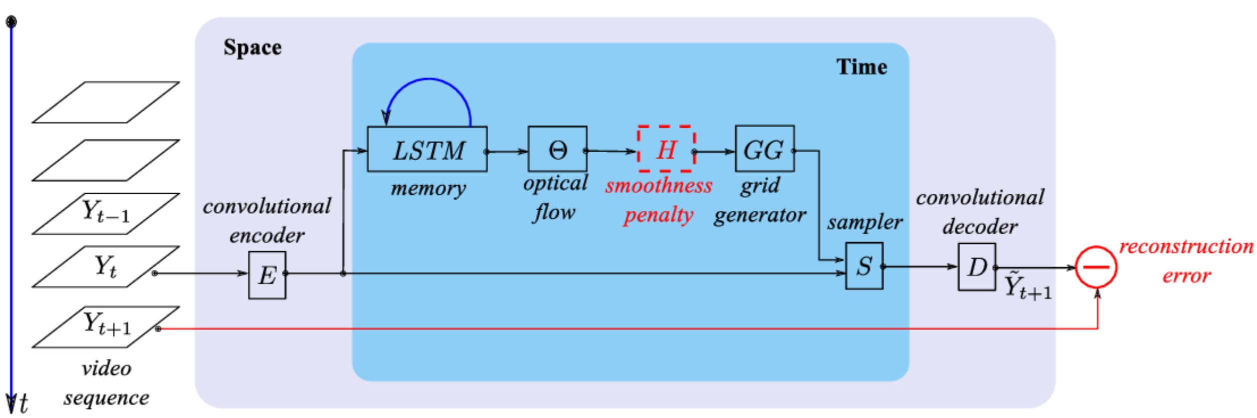

Figure 2: Process of Spatial Convolution

Activation functions are usually used to get the output of node. In other words, activation function is also called as Transfer function. The main purpose of using activation functions is to determine the output of the machine learning model which we have developed using neural networks. With respect to the results of the activation methods, we will be able to extract the output. The activation function is majorly of two types, one is Linear activation function and non-linear activation function. We have applied ReLU and sigmoid activation functions in our machine learning algorithm to decide on whether any abnormal events casted in the input video clip.

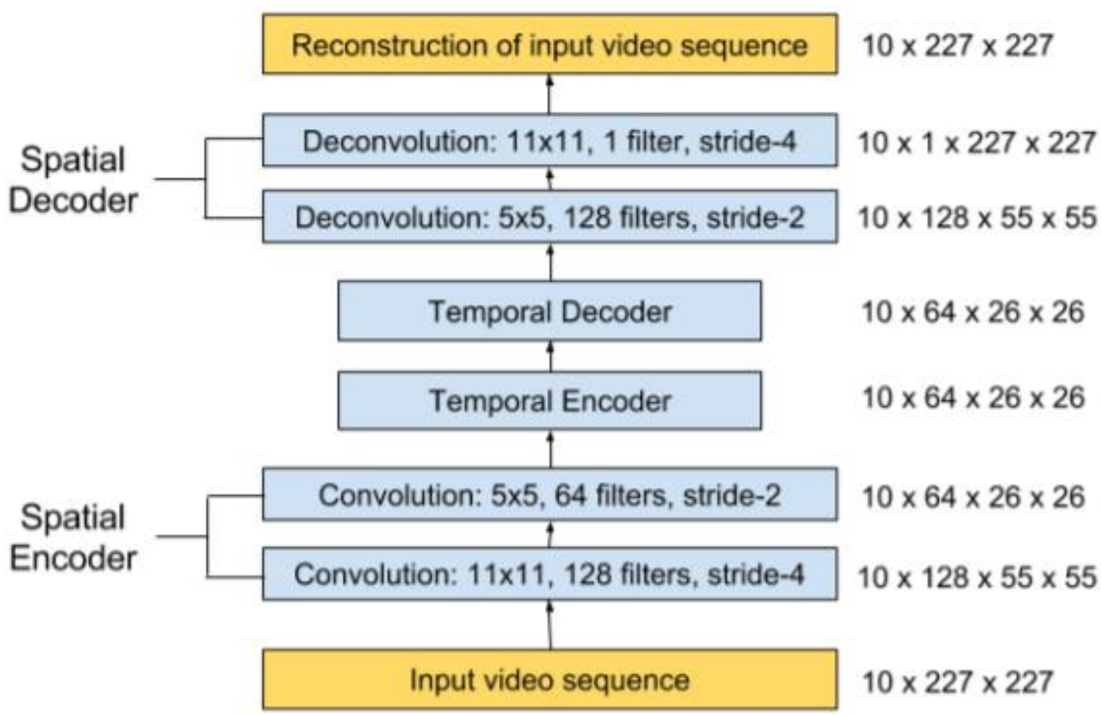

Figure 3: Spatial Convolution - Architecture

This is the overall architecture of the spatial convolutional network. Here, T is taken as a sequence of length that is passed as an input which is reconstructed as an output from the input sequence. The size of each output layer obtained is denoted in the right side of the image. Only one frame at a time is taken as an input by the spatial encoder. Once all the frames have been processed by the encoder, the features are further concatenated and passed as an input to temporal encoder that captures the motion encoding. The work of the decoders is to mirror the encoded features that is finally used to reconstruct the video.

CONVOLUTIONAL-LSTM:

Convolutional Long Short-term memory(ConvLSTM) is a model that is specifically used for predictions on the video frames. When compared to the typically fully connected Long Short-term Memory algorithm, In ConvLSTM the operations on the matrix are done using the convolutional layers. ConvLSTM architecture comes under the category of Recurrent Neural Networks. It follows the same behavior of Long Short-term memory, which is passing the state of the model generated from the previous to the next model layer in the list. The same has been followed with the sequence of images passed to the ConvLSTM Layer. Usually, the layers handle and pass the 1-dimensional vector data in the Long Short-term memory model, but with the ConvLSTM architecture, the model will be capable enough to handle 3-dimensional data which may be a list of images or video frames divided into a sequence of image. Overall, when we need to analyze something with the sequences of video frames, the combination of Convolutional and Long Short-term memory model - ConvLSTM proven to be the best-fit algorithm.

Keras supports the implementation of the ConvLSMT layer. The Keras will define the data format for the images. It also works at the channel level of the images. So, it is very important that the algorithm holds the data which has been processed in the network layers, as the neural networks make any decision not alone with the results of the recent layer, it makes the decision by comparing and contrasting the results of the previous layers as well. It is very important that the algorithm is capable to hold the data as well as the order of the same. Convolutional-LSTM model (ConvLSTM) has a different approach in performing the data transfer between the convolutional layers. The layer produces the result as a 1-dimensional array and the array is made of the features obtained from the image which got processed on the layer.

Rectified Linear Unit (ReLU) activation and Sigmoid has been used has an optimizer to perform the optimization on the fully connected layer. Required smoothing functions are applied for the best performance. We create an instance of the bag with the gathered video clips by processing and dividing a video clip in to thirty-two non-overlapping segments, and the segment count is set randomly. Also, it’s worth mentioning that we have applied multi-scale overlapping in processing the video segments as a part of experimentation and unfortunately, it doesn't have any effect on the accuracy of the abnormal event detector. We fetch bags of both the positive and negative segments from the processed bags as a batch and compute the functionality from the selected bags. We compute gradients for the selected features/variables and obtain the computation graph by applying the chain rule.

Adding up detailed information on ReLU, it stands for rectified linear unit, and it comes under the category of an activation function. In neural networks, ReLU is considered the most preferred activation method or function, especially in Convolutional neural networks. Also, in cases when the machine learning engineer is not sure about what kind of activation method to be used in a case, then ReLU is the better choice. It’s essential to know the working mechanism of ReLU in terms of comparing the values. It is linear or identity value of all positive values and 0 of all negative values. ReLU is considered most as it takes very less time to train the model and run the same. Also, it doesn't have any complicated mathematical operations to perform the comparison. The converges rate of ReLU is also faster when compared to other activation functions. It doesn't have gradient problem as well, which is very usual in other activation functions like tanh and sigmoid.

4.5 EVALUATION

The performance and accuracy of any machine learning model need to be evaluated and investigate the possible options of improving the same if it’s getting lacked. For our thesis work, we have applied receiver operating characteristic (ROC) curve for the video image frames for analyzing the accuracy of the model execution and area under the curve (AUC) to analyze the performance of the built model. We are not using any specific error rate denotation as it’s not easy to process lengthy video. By the stated way, we have performed an evaluation of both the performance and accuracy of the machine learning model which we have developed.

4.6 DEPLOYMENT

Deployment is one of the major phases, in which the machine learning model is deployed as a solution in online or cloud environment which is then accessible for the public to access the model with a user interface. Software deployment also includes preparing an environment to run the software application which we have developed to solve our business requirement. In our case, it’s a machine learning model which we are going to deploy in a specific environment which will be accessible.

The software deployment starts with performing the software installations on the server which will be required to execute the code modules, then we configuration is one of the major steps in which we customize and configure the base software according to our application requirement, then proceed with testing the application to have a check on whether our application which we have developed in our local system is working in the same way in the deployed environment without any additional errors due to the environmental setup. On top of that, any software model requires to optimize the performance of the application, as it is one of the main parameters which will have an influence over the usage of the software system. In our case, we have not stepped into to performance optimization phrase due to time constraint for the development activity. Also, no major consideration on the manual or automation testing, which we have considered as out of the scope for the thesis work.

We have deployed our machine learning model with the user interface in an environment called PythonAnywhere. PythonAnywhere is an integrated development environment which is accessible over online and it also provides web hosting services in which we can deploy our code modules. It provides command-line interface by creating a new console instance window online, which shall be used for any library and package installations. PythonAnywhere is usually used for deploying the python code as it is completely free for a trial period, on the other hand, the integrated development environment is user friendly as well.

The first step to deploy the code in PythonAnywhere is we need to push the code from local development system to code repo like GitHub, the reason behind we push the code to GitHub is, we use code cloning mechanism in PythonAnywhere to pull the code modules into the deployment environment. Once we are done with pulling the code modules, we proceed with setting up the virtual environment which will load the required base packages to execute/run the software system. As a part of the next step, we shall proceed with installing the requirements.txt file which will have the complete list of libraries required for this machine learning model. If in case, we are not handing the installation with requirements.txt file, we may end-up in problem in knowing and installation the right version of the packages when we deploy or move the code modules to a different environment. Coming back to virtual environment concept, its mandatory to active the environment before we run the code modules. The virtual environment is a container to hold the package details and run the application.

CHAPTER 5: DISCUSSION

As a result, a machine learning model that is capable enough to identify the explosions in the video broadcast has been accomplished. Here we going to discuss the possible results which we can expect from the model to result. Starting with the test case 01, we have uploaded the input video clip which has no abnormal activities happened in the video clip in the user interface which we have developed using the HTML and Python flask web framework for feeding and testing the input with the model. In the mentioned case, the video input will be submitted to the model, then the video clip will be converted into image frames, we also display the frame count for the given input video, as a next step, the machine learning model will get initiate to start processing the video frames. Once the frames got processed, the output will get generated in the form of GIF, or Graphic Interchange Format with the animated raster graphics file which will fast forward the processed video frames and correspondingly will show the output down below the processed frames in a chart representation. If no changes are cast in the chart graph, then it means no abnormal activities identified. Moving on to the next scenario, we input the video clip which has no major impact due to the explosions, in this case, our machine learning algorithm was not efficient enough to identify the explosion due to the low impact level recorded in the video.

To test the actual usage of the developed machine learning model, for this test, we have given the input video clip with a major explosion/bomb blast occurred at one point of time in the total video clip. The process remains the same, the video clip is divided into video frames and processed. As a result, the Graphics Interchange Format image will have resulted, and the graphic shows both the processed frames along with the effects identified in the video. The chart in the result shows the drastic changes in the graph when it detects the explosion in the video. The graph also contains a numeric value in the X-axis and Y-axis which denotes the level of impact occurred in the video clip. The graph value get fluctuates with respect to the occurrence, and would like to mention it's not always the same due to some differences in processing the sequence of video frames.

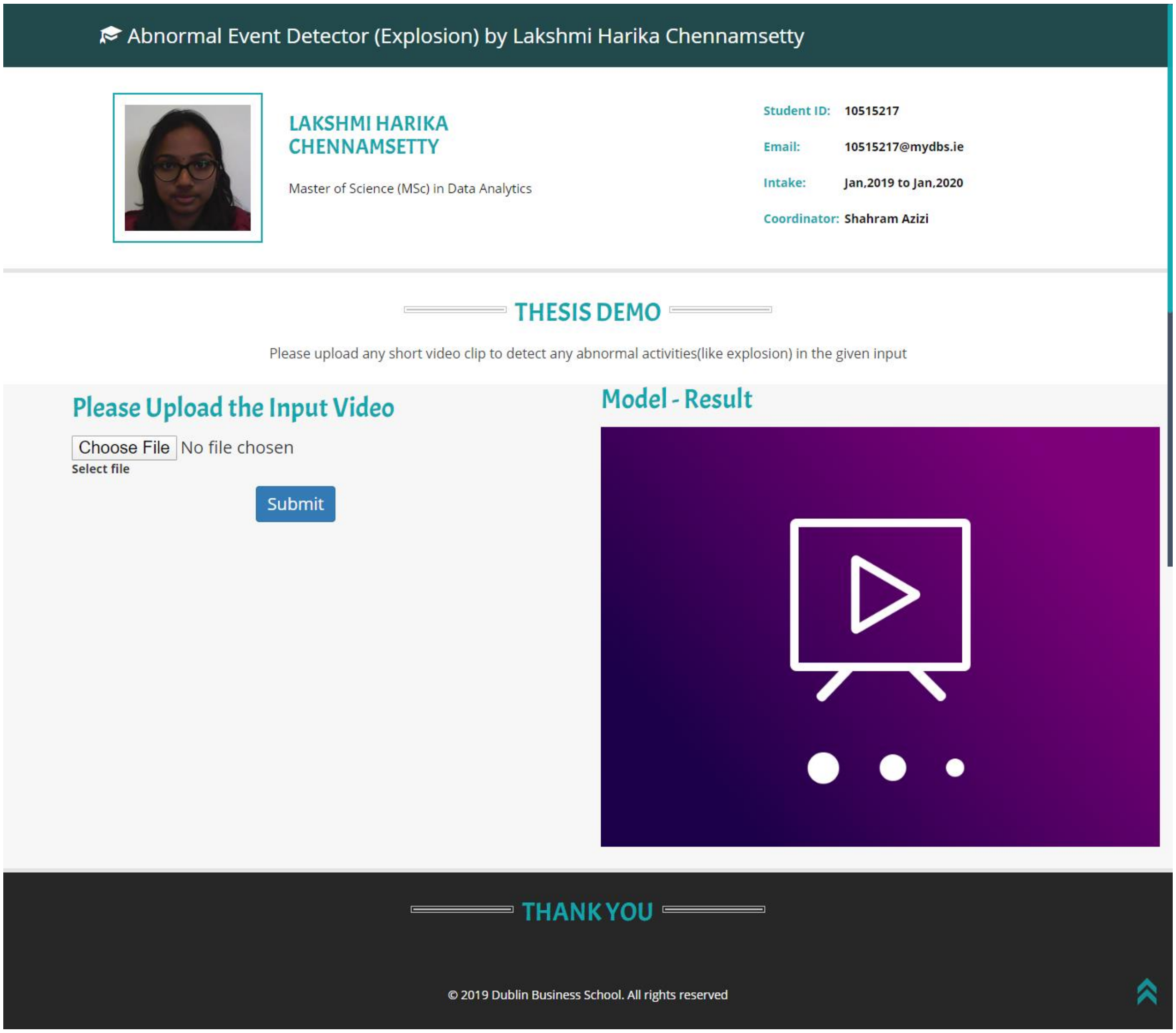

Figure 4: Web Application Interface

The given screenshot 01 shows the user interface of the application which got connected with the developed machine learning model. On the left-hand window, we have given to provision to upload the video clip as an input, and once you browsed the video, upload the same to submit it to server using the submit button. On the other side of the web page, we have given placeholder image, which will actually get replaced with the output gif animated image generated from the model.

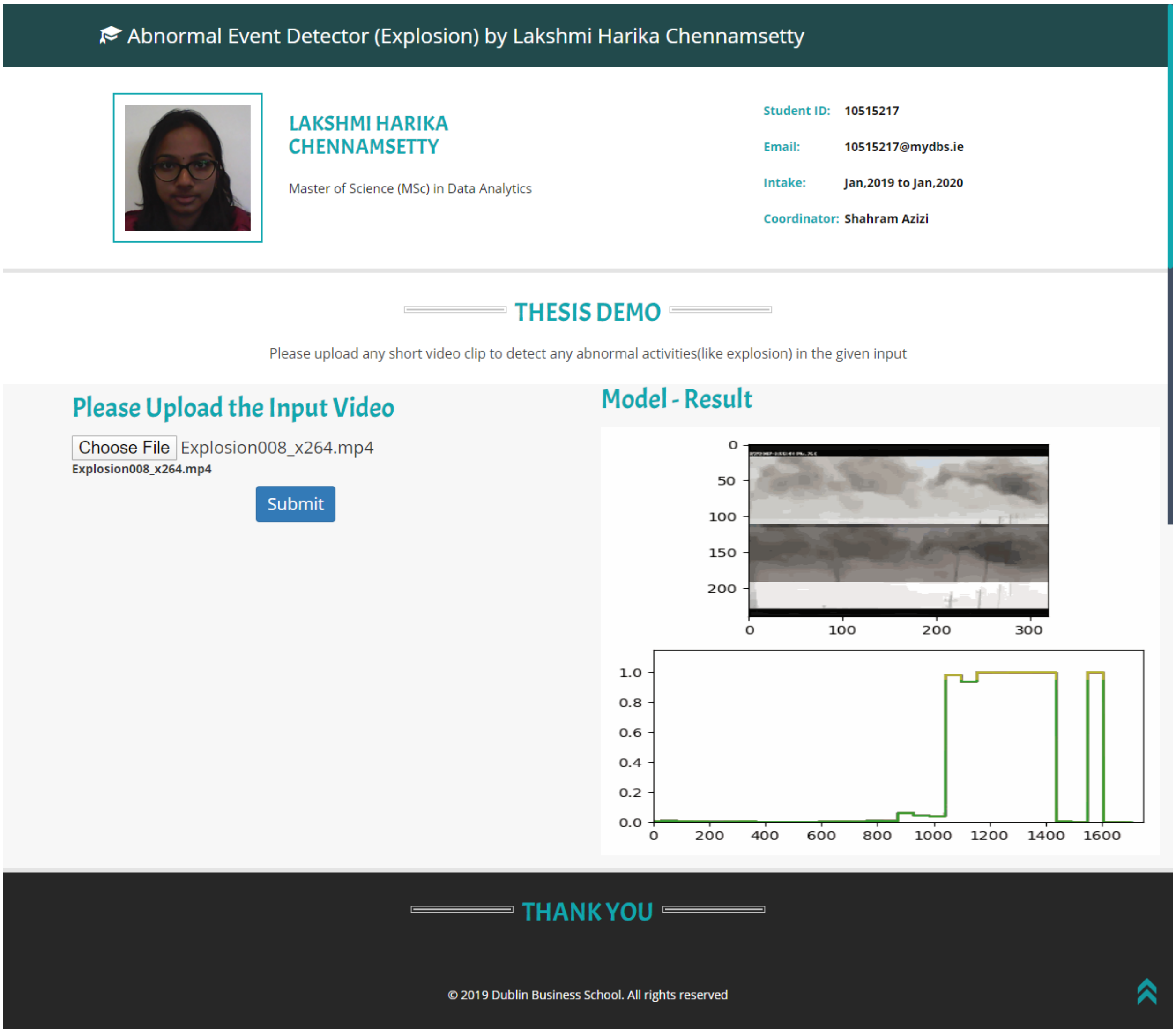

Figure 5:Output of the processed video

Screenshot 02 shows the output for which the input was a video with the explosion. The explosion has created a major impact on the screen for which the camera was in focus. The impact has been recorded in the output graph in the form of the chart on the right-hand side of the web page. In this test, the machine learning model which we have trained to identify the explosions has provided a justified result by identifying the same. As we have mentioned earlier in the document, the model should still be enhanced to identify even if the impact is less.

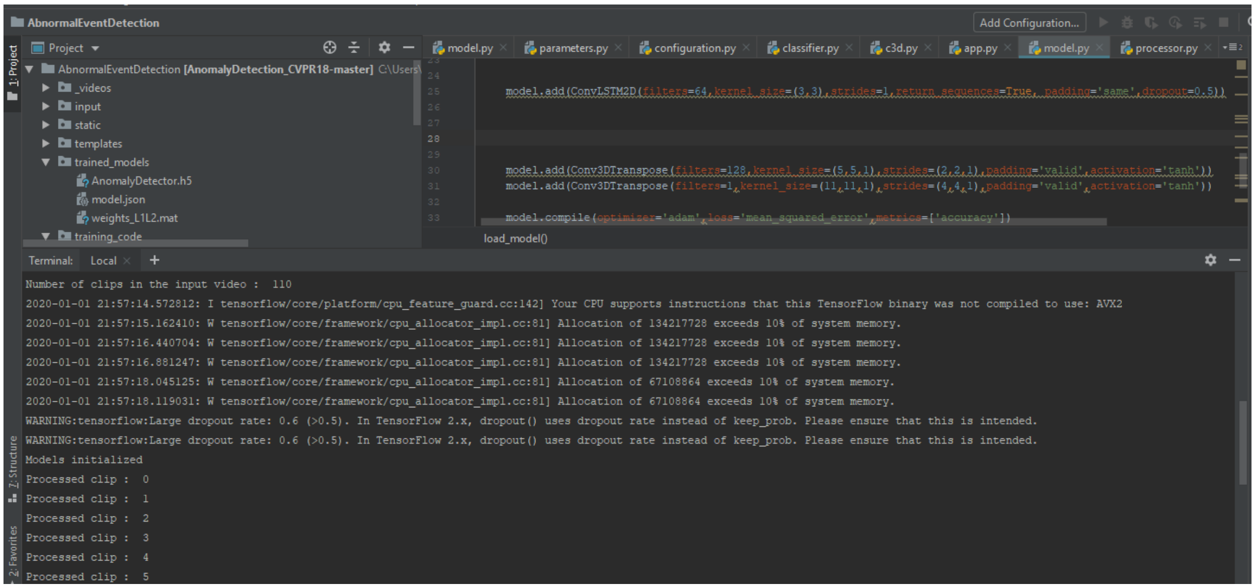

Figure 6:Command line output of the execution

Screenshot 03 shows the console window of the model execution starting from displaying the number of clips in the input video, initiating the model, log of allocating the memory in the CPU to run the model and also the status of the clips getting processed.

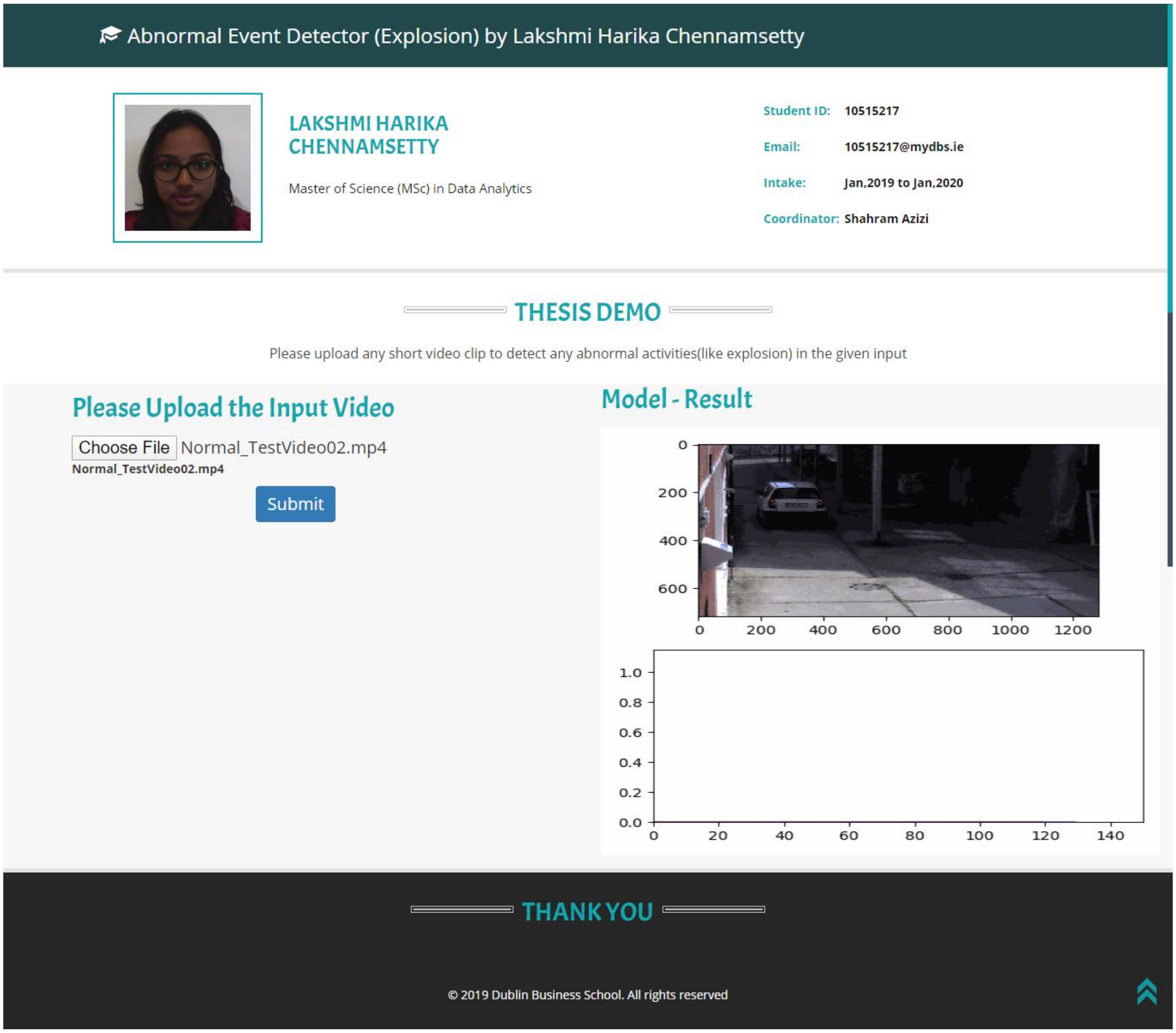

Figure 7:Input video clip without any abnormal activities

In screenshot 04, we have given the input video clip which doesn't hold any abnormal activities in the clip, the video was just a car passing over the road way. In this case, no special actions need to be taken, and therefore detecting any events from the video is not going to help in any way in the real-time use cases. If we check the graph recorded in the output, there was no changes in the line which is getting plotted and it was always in the ground which shows that no abnormal activities has been detected.

CHAPTER 6: CONCLUSION & FUTURE WORK

In this chapter, we have mentioned the conclusion of our thesis work and the future works that shall be taken to enhance the developed model. The long process of developing a machine learning algorithm which is capable enough to identify the abnormal activities as explosions has ended with this chapter. The conclusion part also states the learning we have gained in developing this machine learning model. On top of that, the future works have given a clear route path of what should be done in the subsequent phases if in case we have decided to enhance the research work for making it has a solution that works to solve real-world problems.

6.1 CONCLUSION

The thesis works helped us to understand the approach required to build any machine learning model, starting from the analysis phase, understanding the need from the real-world use cases with respect to the chosen machine learning concept for the implementation, in-depth technical knowledge required to develop the machine learning model, preparing the dataset for training the model, understanding and identifying the features that are required for training the model, choosing the right algorithm for preparing the model, evaluating the machine learning model to analyze its performance and accuracy, and at last also learned on how to develop a web application in which user shall interact with the machine learning model.

Adhering to the process of the execution methodology like CRISP-DM helped not only to develop the solution but also in a proper way so that the gained knowledge can be applied when we are into an organization and as a team member in the development unit. To be more specific about the experience which we have gained in learning and applying the machine learning algorithm, from the summary of the problem statement, We were able to relate the problem with a machine learning algorithm that shall be applied for the chosen case. In our case, we have chosen LSTM (long short-term memory) which is a recurrent neural network (RNN) architecture, as it requires the output from the previous unit to take any decisions.

6.2 FUTURE WORK

For any application we develop, releasing an update is required in a frequent manner, so it is mandatory to discuss and plan about how the current work can be improved and enhanced to a further level by releasing new versions of the same. For this work, we shall plan for many phases and mentioned the direction of future work in this part of the document.

As of now, the model is trained and capable enough to identify only the explosions or blasts which have the major impact in the video broadcast, and this shall be improved to identify the explosions even if it is not creating any major impact in the surveillance area. Moving on to the next phase, the model shall be trained additionally with different other abnormal activities like traffic rules violation, earthquake, burglaries, etc. By enhancing the model in the described way, helps to deploy the model as a product in the public places to overcome the problem of involving additional human resources for surveillance monitoring.

CHAPTER 7: REFERENCES

Agrawal, S., Agrawal, J., 2015. Survey on Anomaly Detection using Data Mining Techniques. Procedia Comput. Sci. 60, 708–713. https://doi.org/10.1016/j.procs.2015.08.220

Amer, M., Goldstein, M., Abdennadher, S., 2013. Enhancing one-class support vector machines for unsupervised anomaly detection, in: Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description - ODD ’13. Presented at the the ACM SIGKDD Workshop, ACM Press, Chicago, Illinois, pp. 8–15. https://doi.org/10.1145/2500853.2500857

Aneetha, A.S., Bose, D.S., 2012. HE COMBINED APPROACH FOR ANOMALY DETECTION USING NEURAL NETWORKS AND CLUSTERING TECHNIQUES.

Chong, Y.S., Tay, Y.H., n.d. Modeling Representation of Videos for Anomaly Detection using Deep Learning: A Review 4.

Devaraju, S., Ramakrishnan, S., n.d. Detection of Accuracy for Intrusion Detection System Using Neural Network Classifier.

Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A.K., Davis, L.S., 2016. Learning Temporal Regularity in Video Sequences, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, pp. 733–742. https://doi.org/10.1109/CVPR.2016.86

Ho, B., n.d. IMAGE & VIDEO RECOGNITION. N. Y. 11.

Luo, W., Liu, W., Gao, S., 2017. A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework, in: 2017 IEEE International Conference on Computer Vision (ICCV). Presented at the 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, Venice, pp. 341–349. https://doi.org/10.1109/ICCV.2017.45

Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., Shroff, G., 2016. LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection. ArXiv160700148 Cs Stat.

Pradhan, M., Pradhan, S.K., Sahu, S.K., n.d. ANOMALY DETECTION USING ARTIFICIAL NEURAL NETWORK. Int. J. Eng. Sci. 2, 9.

Vu, H., Nguyen, T.D., Travers, A., Venkatesh, S., Phung, D., 2017. Energy-Based Localized Anomaly Detection in Video Surveillance, in: Kim, J., Shim, K., Cao, L., Lee, J.-G., Lin, X., Moon, Y.-S. (Eds.), Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 641–653. https://doi.org/10.1007/978-3-319-57454-7_50

Xu, D., Ricci, E., Yan, Y., Song, J., Sebe, N., 2015. Learning Deep Representations of Appearance and Motion for Anomalous Event Detection. ArXiv151001553 Cs.