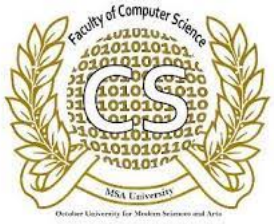

October University for Modern Science and arts

Faculty of Computer Science

Graduation Project

Title: Crime Detection Using Deep learning

Supervisor: Dr Ahmed Farouk

Name: Kareem Ahmed Saad El-Mallahy

ID: 195363

Abstract

CCTV cameras are installed all around the world in each country and each city but still the crime rate is really increasing rapidly as we do not have the manpower to monitor the all the events and places, so we can implement an intelligent crime detection system using deep learning. We suggest utilizing both regular and abnormal videos to learn anomalies. We will be using a crime dataset which is called the UCF-crime dataset. This dataset consists of 128-hour video dataset that is the first of its type on a wide scale. It includes 1900 lengthy and uncut real-world surveillance footage, as well as 13 actual abnormalities such as fights, car accidents, burglaries, robberies, and other routine events. The goal of this project is to be able to detect the crime happening without involving any human in the process and raise an alarm to the police to fasten the process.

Table of Contents

Table of Figures

Chapter 1: Introduction

1.1 Introduction:

Crime is an act that is committed by a person who is seen to be a criminal and that may result in the bodily hurt or loss of property or damage to another person. According to the seriousness of the crime, the country's authorities always penalise the individual who committed this conduct. There is a sizable quantity of crime in every society. Almost everyone is affected in some way by its expenses and effects. Costs and effects can take many different forms and dimensions. Additionally, some expenses come and go while others last a lifetime. Death is, of course, the ultimate cost. Other costs that victims can face include medical bills, property damage, and missed pay. The three separate components of crime detection are the discovery of a crime, the identification of a suspect, and the collection of sufficient evidence to bring the suspect before a court (fig.1). The victim of a crime that occurred on January 10, 2017, was identified as Donald Coty Jr. He was found shot and killed instantly in the driver's seat; the crime took place on 12 and Paseo [4], and if a crime detection system had been installed on any of the street cameras at the scene, we might have been able to save him or catch the shooter. Our objective is to aid in the criminal's capture and stop his escape in order to make society a safer place to live. We also hope to inform police officers of the crime as it is happening in order to speed up the investigation and avoid having to wait for someone to call the police before the criminal can be apprehended. The current strategy to detecting crimes is to use a variety of computer vision algorithms, Bayesian techniques, artificial neural networks, spiking neural networks, and fuzzy logical procedures [1]. The Bayesian methods has some pros such as:

1. Capacity to incorporate results from placebo-controlled trials and direct (from head-to-head trials) trials into a single analysis.
2. Capable of delivering outcomes for any pertinent comparison inside a network of links.

And has some cons such as:

1. Due to the fact that the studies are generally performed utilising unfamiliar or foreign software, many investigators find them to be difficult to interpret (usually run using WinBUGS).

2. Compared to certain other techniques, it requires a greater level of statistical expertise [5]

Figure 1 Crime Detection

1.2 Problem Statement

Building an intelligent crime detection system is very difficult due to the very high computing requirements as an intelligent crime detection system be able to learn from experience, the system need to be secure and to be able to adapt according to current data. Also, to track all places and the all the events the authorities of the countries need a huge manpower to do this. Lastly there is some extrinsic problems as noise, shadows and low resolution, real-world videos are tough to work with.

1.3 objective

The project should present a fully automatic system to detect any crimes happening without involving any humans in detecting the crime. The system should understand that there is a crime happening to send all the needed resources to the crime scene. The system should raise an alarm that there is a crime happening now.

1.3 Motivation

In today's environment, an automated system for detecting crime is a must as it is important for the law enforcement and the people. The method would aid in crime reduction by making detection simple and instantaneous [1]. Also, with the increasing rise of smart cities, crime detection systems are being integrated to increase security [3].

1.4 Thesis

In this thesis the first chapter will be providing an introduction about the project, aim and motivation. Moreover, the second chapter will be providing a background and a literature review of the previous work in the same area of this research. Furthermore, the third chapter will be providing the materials which is data, tools, and environment and the third chapter will provide the methods that will be implemented in this project.

Chapter 2: Background and Literature Review

2.1 Background

Crime detections consist of three distinct aspects that are the discovery of the crime, depict of the suspect and collecting evidence to indict the person in front of the court. As a result of security concerns, the number of surveillance cameras is rapidly increasing by creating an automated system that can detect the crimes without the involving of the human this method will be able to reduce the number of crimes as crime detection system will make the detection easier and instantaneous. Many researchers tried to make an automated crime detection system before, but they only trying to detect the weapons and there are some crimes that does not involve any weapons. Most of researchers use a pre-trained models rather than building a custom model. So, even if the Crime detection system made before gives a high accuracy still it does not work as it should be because the model only detects the weapons. There were some algorithms that has been used in this field before for example the Deep learning algorithm and K-Nearest Neighbors algorithm (KNN).

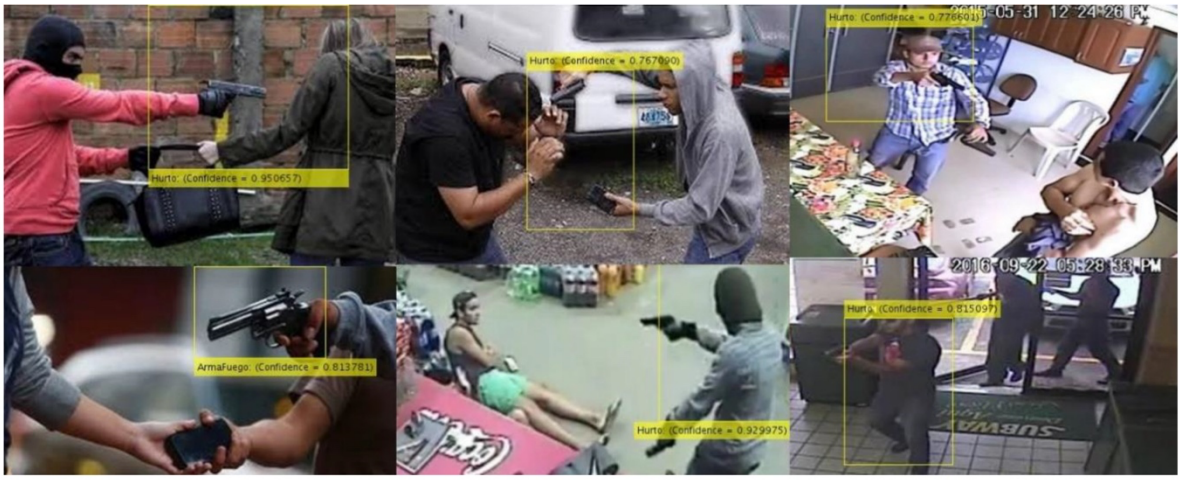

Figure 2 Weapon Detection

2.1.1 Previous algorithms

Deep Learning: Deep learning is a branch of machine learning. Machine learning techniques like deep learning learn traits and tasks from data. Data samples include pictures, text files, and audio files [12].

K-Nearest Neighbors algorithm (KNN): One of the most fundamental Machine Learning algorithms is the KNN technique, which is based on the Supervised Learning methodology. The KNN approach places the new case in the category that resembles the

current categories the most on the assumption that the new case/data and previous cases are comparable. A new data point is classified using the KNN algorithm depending on how similar it is to previously classified data, which stores all accessible data. This suggests that the KNN approach [13] may swiftly sort new data into a well defined category.

2.2 Previous Work

2.2.1 Research 1: Factex: A Practical Approach to Crime Detection

2.2.1.1 Strategy and structure:

All contemporary cities are currently coping with a significant problem called road crime. Road travel is a popular form of emigration for criminals. Thefts and many other crimes go unreported and unsolved because there isn't enough evidence. The primary goal of this research is to identify criminals who are fleeing in automobiles by checking the licence plates of the vehicles against a database or by checking the faces of people on the street against a database of criminal faces, as demonstrated in (fig.2). The OCR system was employed in this study (Optical Character Recognition). The proposed system uses the K-Nearest Neighbors algorithm (KNN) for text recognition and the HAAR Faces Classifier and SVM for face identification. [6]

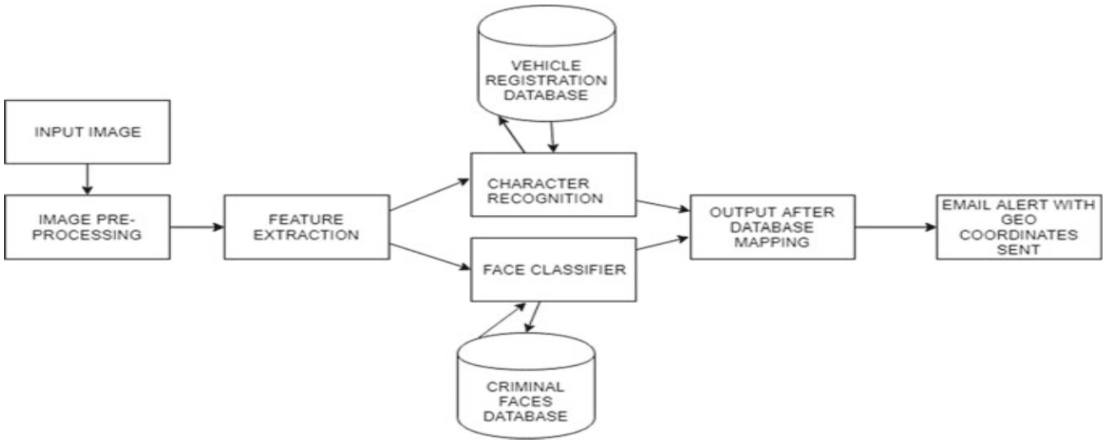

Figure 3 System Architecture [6]

2.2.1.2 Data

The dataset was made by the researchers, however it has a problem in that there are only 50 photos in total, each of which has a facial expression [6].
The dataset consists of images that a single person's camera took. Despite being a small dataset, it may nevertheless be used as a benchmark for improved monitoring.

2.2.1.3 Method evaluation

The OCR system can read licence plates with a 98–99 percent accuracy rate, making it extremely accurate [8]. However, the OCR still takes a lot of time and doesn't use face recognition to find the criminal [6]. The KNN algorithm is among the simplest and provides a high level of accuracy. However, KNN use a lot of memory, and the output is affected by noise and irrelevant properties.

2.2.1.4 Results evaluation

The system for detecting crime has already been improved, yielding results that are far more accurate and efficient. It uses facial recognition technology as well as text recognition technology. Regardless of noise, intensity, or other factors, the KNN approach employs a Gaussian Blur filter to handle all types of images for number plate recognition. Similar to this, the self-built database makes considerably more accurate predictions about how lighting and human movement would affect face recognition. The recommended method operates with an accuracy of much more than 85% accurate detections under ambient light conditions.

2.2.2 Research 2: Crime Intention Detection System Using Deep Learning

2.2.2.1 Strategy and structure

In this study, it was highlighted how commonplace CCTV is across the globe, but that these cameras do little to prevent crime; rather, they only keep track of it so that police officers may utilise it when someone reports a crime to them. Pre-trained deep learning models were applied. The main objectives of the research are to make CNN run without performance degradation with less training data, identify weapons in less time with better accurate conclusions and fewer false positives than machine learning technologies. Pre-trained models, like GoogleNet and VGGNet-19, have learned from millions of pictures and can accurately identify objects in new

images, as demonstrated in (fig.3). Due to the VGGNet19 model's superior training accuracy, they selected it. It recognises and classes things appropriately. The input layer, which does pre-processing, accepts input frames. The Convolution, Max-pooling, and FC layers are then given the preprocessed pictures to conduct feature extraction, filtering, mapping, and classification. If any criminal intents are found, the output layer delivers a security message about crime intentions using a registered API [8].

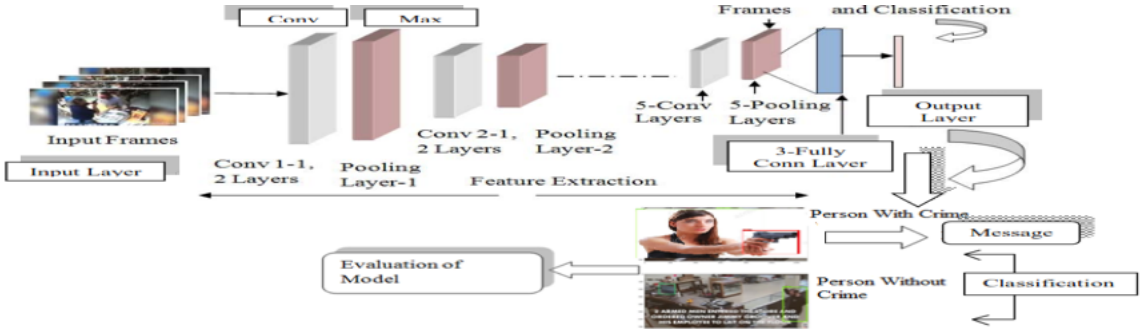

Figure 4 System Architecture[8]

2.2.2.2 Data

Datasets for both videos and photos that were gathered from Google and YouTube were used to test the developed system. The statistics gathered concern crimes such as robbery, murder, and other unlawful acts committed while carrying a weapon, all of which are forbidden in public locations like ATMs, banks, and other gathering places [9].

2.2.2.3 Method evaluation

Convolution Neural Networks (CNN), a deep learning technique, GoogleNet, and VGGNet-19, some pre-trained models, were employed in this study. CNN offers incredibly high accuracy in picture identification tasks. CNN does not, however, encode an object's position or orientation. Furthermore, a substantial amount of training data is required [9].

2.2.2.4 Results evaluation

Both the pre-trained models used for this research, VGGNet-19 and GoogleNet, provided differing levels of accuracy. VGGNet-19 produced better results,

with an average accuracy of 92%. Additionally, the GoogleNet model provided average accuracy of 69%. Additionally, GoogleNet required more processing time than VGGNet-19.

2.2.3 Research 3: Design of an intelligent video surveillance system for crime prevention: applying deep learning technology

2.2.3.1 Strategy and structure

This study aims to develop a smart surveillance system that can constantly monitor in real time without requiring human interaction. Deep learning technology will be utilised to overcome the shortcomings of the current video surveillance system through the data processing model design to display data for crime detection after building an artificial intelligence server and a video surveillance camera. Additionally, this design offers a smart surveillance system that employs real-time processing to send a video image and a notification message to the web in order to quickly and effectively identify crimes. Our recommended technique offers an excellent balance of speed and accuracy in recognising crimes and catastrophes. This proposed concept instantly transmits photographs and notifications to a user's app. This method, in contrast to other recommended ones, enables users to quickly recognise and detect dangers utilising real-time visual data through socket connection. While deep learning in artificial intelligence is continuously generating real-time picture frames, video streaming is also achievable. The system environment includes Python flask for the web, Python TensorFlow for deep learning, and Python socket for the Raspberry Pi. The GPU server has one Tesla p40 GPU with 24G memory and four vCPUs with 30G RAM, as illustrated in (fig.4). The function of raspberry pie is comparable to that of the current video surveillance system. It both captures pictures and transmits the image frame from the camera. The GPU server carries out three tasks: webpage opening, automatic recognition deep learning and notification algorithm, and socket connection with the Raspberry Pi. The GPU server has a thread configuration with each function being made up of three parallel-running threads. Giving the user's application real-time notifications can help prevent crimes more proactively [11].

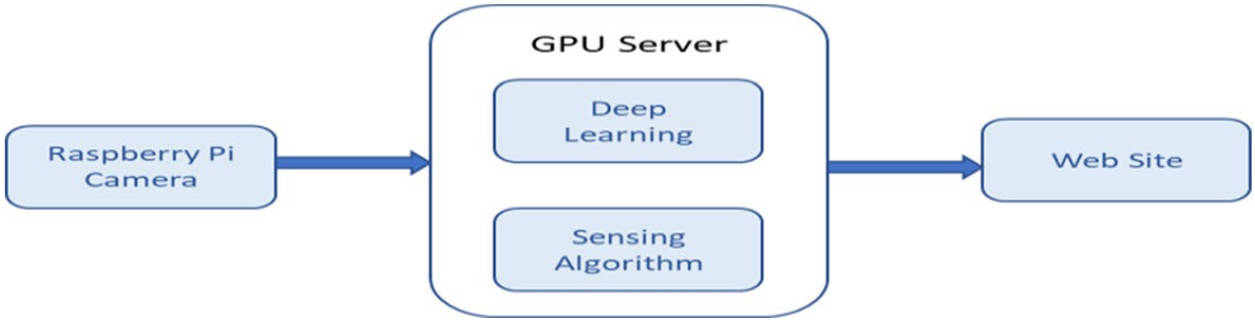

Figure 5 System Architecture[11]

2.2.3.2 Data

The COCO net dataset, a large dataset for item recognition, segmentation, and captioning, was employed in this study. However, the COCO net dataset does not pick up on some things, such deadly weaponry and fires. By labelling additional photographs, recognition must be explicitly taught [11].

2.2.3.3 Method evaluation.

For deep learning, the researcher is employing Python's Tensorflow, which is incredibly scalable. Any work may be completed with Tensorflow. Users may build any kind of system since TensorFlow can be installed on any machine and offers a visual representation of a model [10]. Because the functions created by the system environment are not straightforward, this system is difficult to implement [11]. Additionally, Tensorflow's slow speed means that this solution won't be the fastest.

2.2.3.4 Results evaluation

This study demonstrated a sophisticated monitoring system that looks for and prevents illegal conduct. This suggested model suggests that criminal and catastrophe notifications may be made faster and more correctly if deep learning technologies are applied to the servers linked to the notification system [11].

Chapter 3: Material and Methods

3.1 Materials

3.1.1 Data

In this project we will be using the UCF-Crime Dataset. This dataset consists of 13 types of crime with total number of 1900 video. This dataset is 128-hour dataset. It comprises of extensive, uncut surveillance recordings that cover 13 real-world abnormalities, such as Abuse, Arson, Assault, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. The dataset is divided in 13 folder each folder contains several videos. These anomalies were chosen because they pose a serious threat to public safety. This dataset is downloaded from the internet.

Figure 6 Frames From the Dataset

3.1.2 Tools

We'll be using Anaconda software, one of the most well-liked open-source platforms with more than 1500 python packages that makes it easier to manage and deploy artificial intelligence (AI) and machine learning systems. Python 3 will be used because it is an open-source, free programming language that is simple to read, write, and learn. Because Keras is a high-level neural networks library developed in Python, using it is really simple and easy. A Python programme called Theano makes it feasible to evaluate mathematical operations, such multi-dimensional arrays, efficiently. It is mostly used for creating Deep Learning projects.

3.1.3 Environment

Apple M1 chip 8-core CPU with 4 performances cores and 4 efficiency cores, 8-core GPU and 16- core neural engine. Apple's M1 Chip is faster than the intel processor in learning.

3.2 Method

We will discuss and clarify our solution process as well as the strategy we employed in this part (algorithms)

3.2.1 System Architecture overview

The goal of this research is to detect if there is crime happening not only by detecting objects or weapons. Also, by detecting the behavior happening by the criminal in the video. We will be using Multiple Instance Learning (MIL) which is a type of the supervised learning. Furthermore, instead of receiving individually labeled instances we receive a set of labeled bags and each bag contain many instances. For example, if the video contains only one frame that is abnormal then the whole video will be labeled as a positive bag (Contain crime) but, if the video does not contain any abnormal frame, then the video will be labeled as a negative bag (Does not contain crime). Then we extract the features of the video segments using a pre-trained 3D Convolution Network after this we train a fully connected neural network. lastly, we will the check the accuracy if we reached an acceptable accuracy.

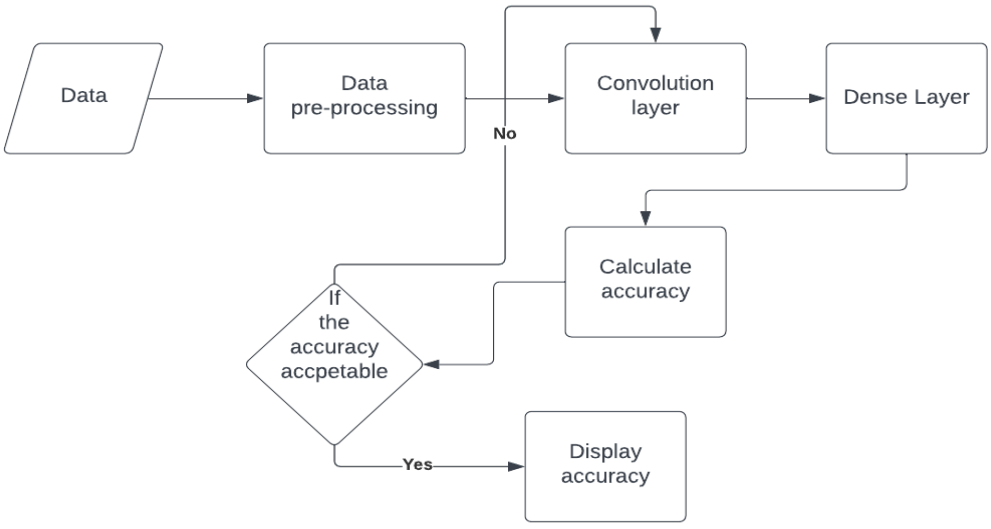

Figure 7 proposed system Architecture

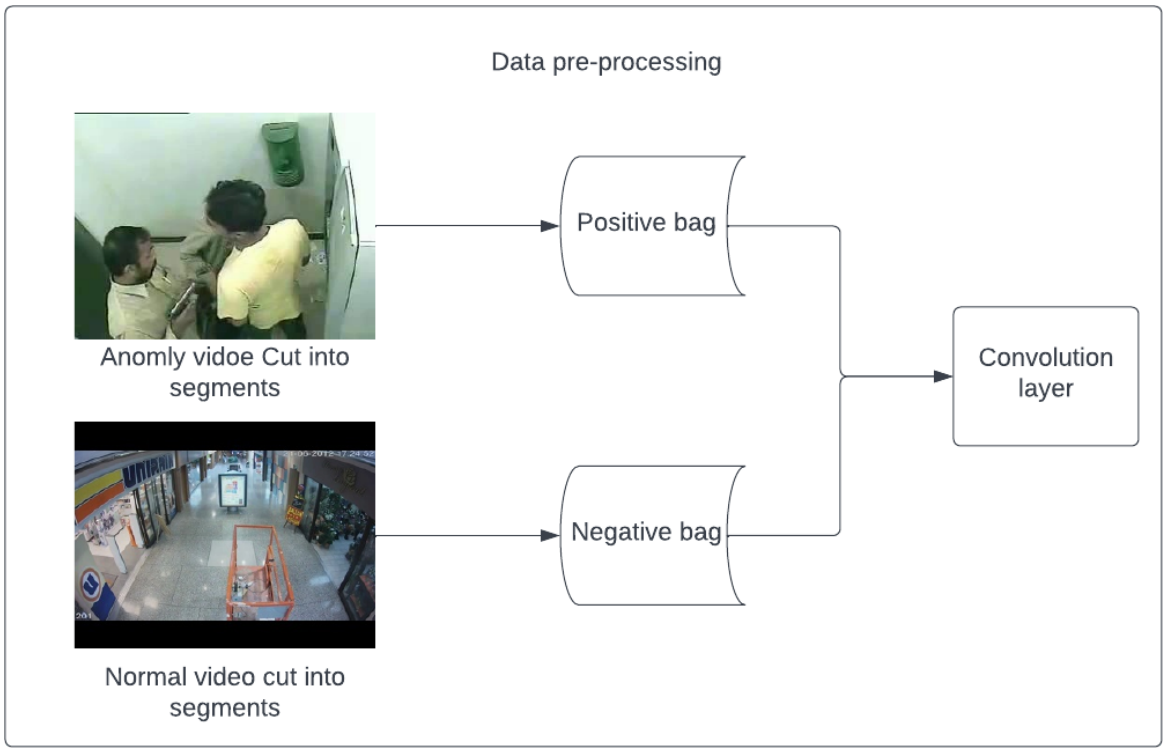

Figure 8 Data preprocessing

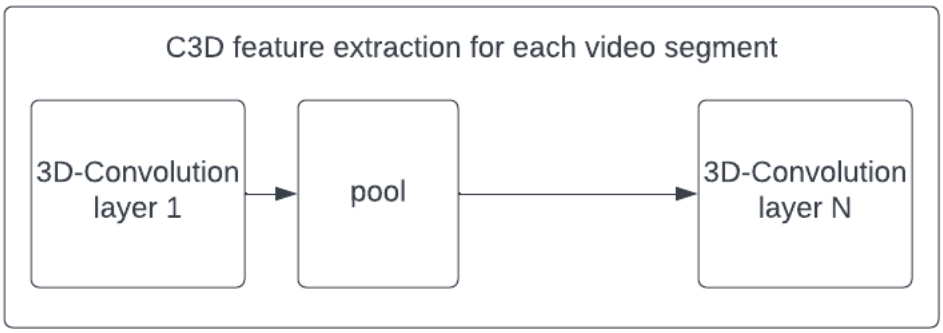

Figure 9 3D Convolution Network

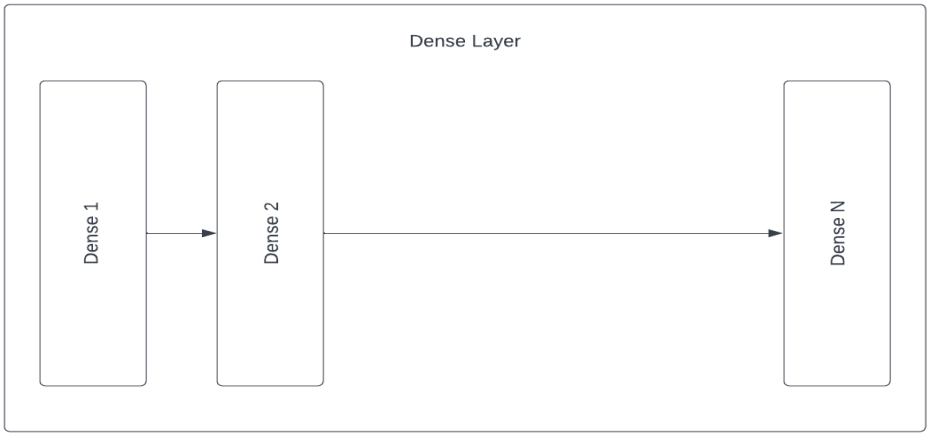

figure 10 Dense Layer

Chapter 4: System Implementation

The technical details of the system's successful implementation will be covered in more detail in this section. On the one hand, the system's technical components and the construction techniques will be evaluated; on the other, a complete analysis of the design choices that were taken in order to achieve the system's goal will be done. We'll discuss how to use the system in practise in the second part. In the second portion, the system's design and the functions played by each of its parts will be thoroughly examined. The last portion of the chapter will then be displayed, showing the inputs and outputs of each system's individual components as well as the outputs that go along with them.

4.1 System Development:

A sequence of modifications that went through several phases finally resulted in the completion of the criminal detecting system. This system's objective is to determine whether or not a crime is depicted in the video. Since the dataset was given in the form of movies, the films were first divided into frames and stored to the computer. Each video received its own folder, including all of its frames. After framing the movies, begin preparing the information to be included to the bags (MIL algorithm). The frames were downsized from their original 320x240 size to 64x64.

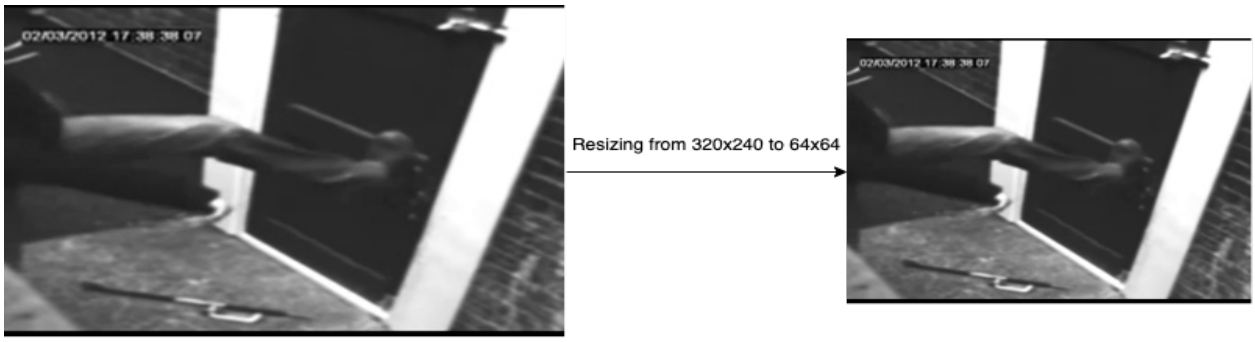

Figure 11 Resizing Frames

When there are 400 frames within a bag, check to see if at least one of them contains a crime; if not, the bag will be classified as normal. This process is known as MIL (Multiple Instance Learning), and it is carried out by adding these frames to bags. This 3D CNN (3D Convolution neural network) was developed using Keras, and a model was designed to determine if it will produce successful results or not. A 2D Long Short-Term Memory (LSTM) model is developed after the 3D CNN is tested. LSTM is used in the area of deep learning. This kind of recurrent

neural network (RNN) can anticipate sequences and learn long-term relationships. Since LSTM includes feedback connections, it can understand both single data points like images and the complete sequence of data. The LSTM is a kind of RNN that excels on a number of different problems. The data was then divided into training, validation, and testing, with training accounting for 60% of the training set, 20% of the validation set, and 20% of the testing set. Then a fully connected layer, followed by an output layer, is coupled with both the 3D CNN and 2D LSTM layers. Several pieces of software, including OpenCV, Tensorflow, Keras, Numpy, Matplotlib, Google Colab, Jupyter Notebooks, and Pycharm, were utilised in the creation of this system. There were 400 frames in each bag, each of which included a label. For the bags that contained crimes, the labels read 1, while for the other bags, 0 was used.

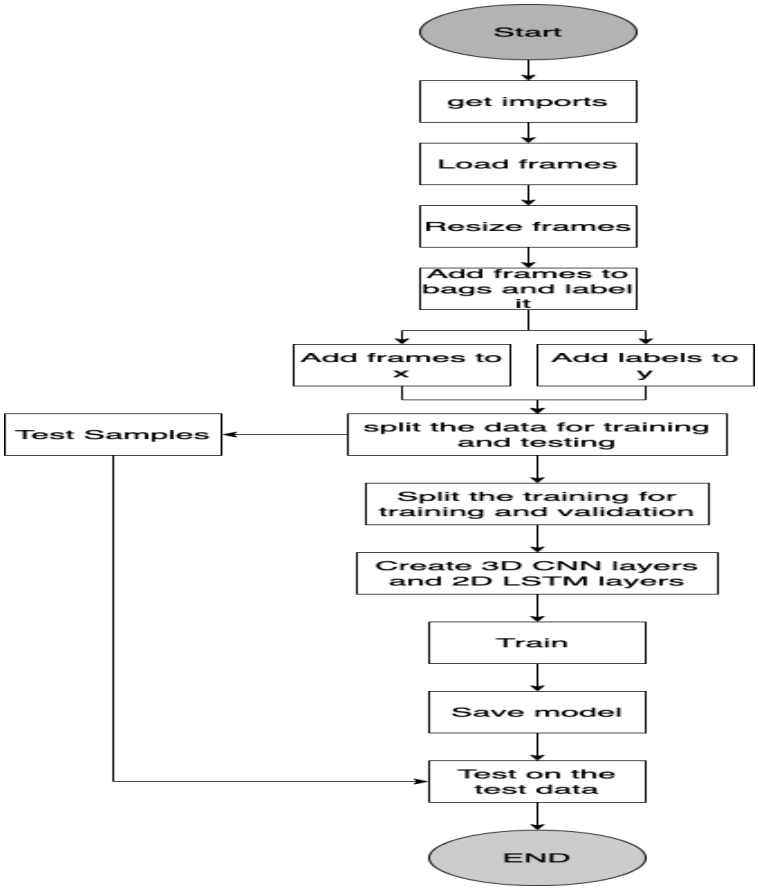

Figure 12 Diagram for system development

4.2 System Structure:

The first part's components provide this information. The first section of this article covers how data is exchanged between the various pieces of the system, which is currently

functioning well. In the second section of the chapter, we will present a tensor-board graph and systems that meet these design specifications, as well as their functioning.

4.2.1 System overview:

The criminal detection system was created by combining the three processes. There are several steps in the initial stage. The first stage is to read the movies. The second is to chop each film into frames and label each frame. The third is to resize each frame from 320x240 to 64x64. The fourth is to add frames to the bags and name each bag, with each bag containing a different 400 frames. Bags were identified by inspecting each frame within the bag; if at least one frame was marked as a crime, the entire bag was marked as a crime. The final phase of this stage is to divide the data into 60 percent training, 20 percent validation, and 20 percent testing if none of the 400 frames had a frame that contained evidence of a crime. If not, the entire bag is then categorised as normal. The preparation stage for data is this one. The training step comes after, when a CNN model and an LSTM model were coupled. The last step in this stage is to determine the training accuracy and the validation accuracy. In this stage, we train our combined model on the bags that were created in the previous stage, where the X will include the bags and the Y will contain the labels. Then, in the last stage, we put our model to the test using the test data we created in the first stage and provide a classification report that includes the precision, recall, f1-score, and support. Next, determine the testing accuracy, and then produce the confusion matrix. The next figure number no. will display the entire system's phases.

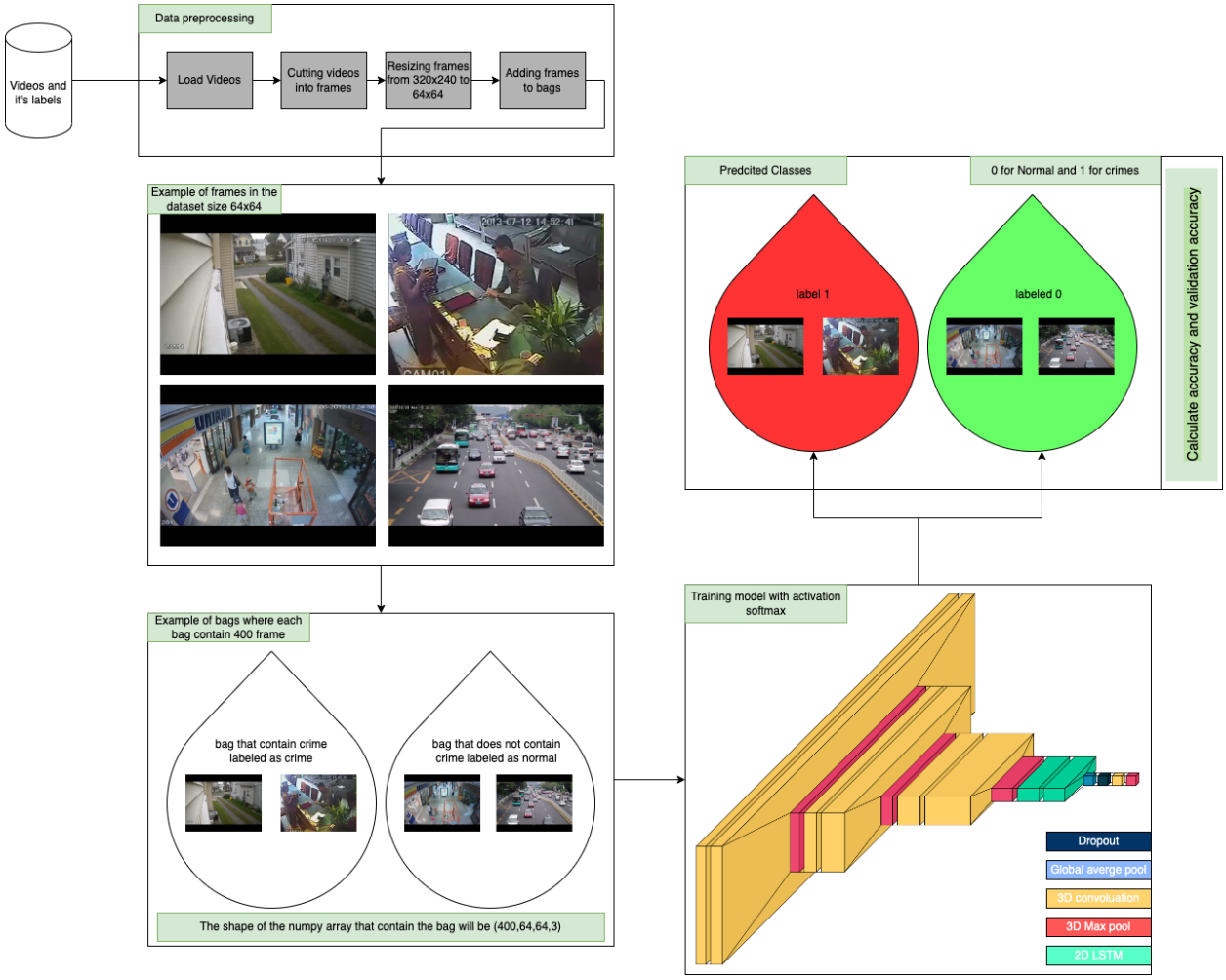

Figure 13 System Overview

4.2.1.1 CNN & LSTM Architecture Overview:

The model was trained on the many iterations so that it can distinguish between recordings of everyday events and films of crimes. Here is a schematic that shows how the two models that are combined in figure no. The 3D CNN begins with an input shape of (none, 400, 64, 64, 3), followed by six 3D convolution layers with the following specifications: kernel size (3x3x3), strides (1x1x1), filters 2, 4, 8, 16, 32, and 64, respectively, and activation function RelU. Following each pair of 3D convolution layers, a 3D max pool layer with a 2x2x2 pool size is applied (2x2x2).

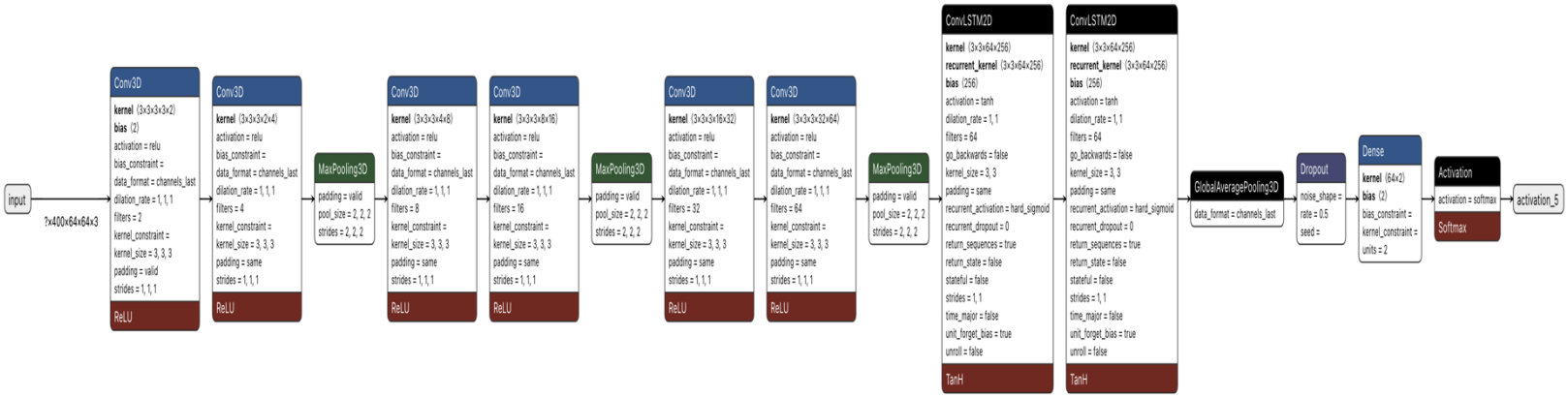

Figure 14 Model Architecture

4.2.2 TensorBoard:

The tensorboard diagram that depicts the whole system architecture is shown in the diagram below.

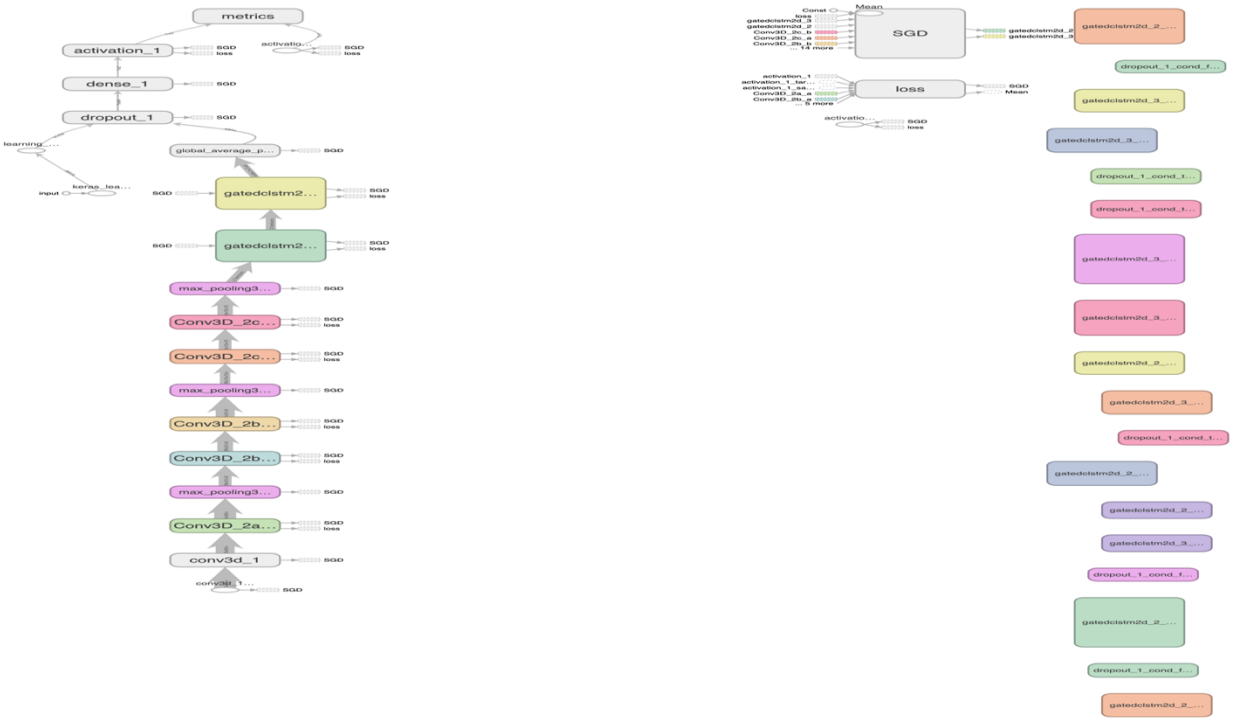

Figure 15 Tensor Board

4.3 System Running:

In this section will talk about the system's input and output in this part, going into further depth about each step.

4.3.1 Data Selection:

We first read the films, then we chop them into frames at a speed of 60 frames per second, with a form and size of 320 x 240 that will eventually be scaled as seen in (figure 16).

Figure 16 Cutting Frames

4.3.2 Data preprocessing:

We retrieved the frames that we had previously cut, which would have given us a size of 320 × 240, and we resized it to 64 x 64. By iterating over all the frames, taking each 400, and then combining them into a bag, we can make the bags. After placing the 400 frames in the bag, look to see whether any of them include criminal activity. If so, the entire bag will be marked as criminal activity. And the bag will be labelled as usual as indicated in case none of the 400 frames contain any that depict a crime (figure 17). Add the bags to the X and the labels to the Y, and then divide the data into training and testing groups of 60% each. And with this, the data preparation is complete. Then the training phase comes next.

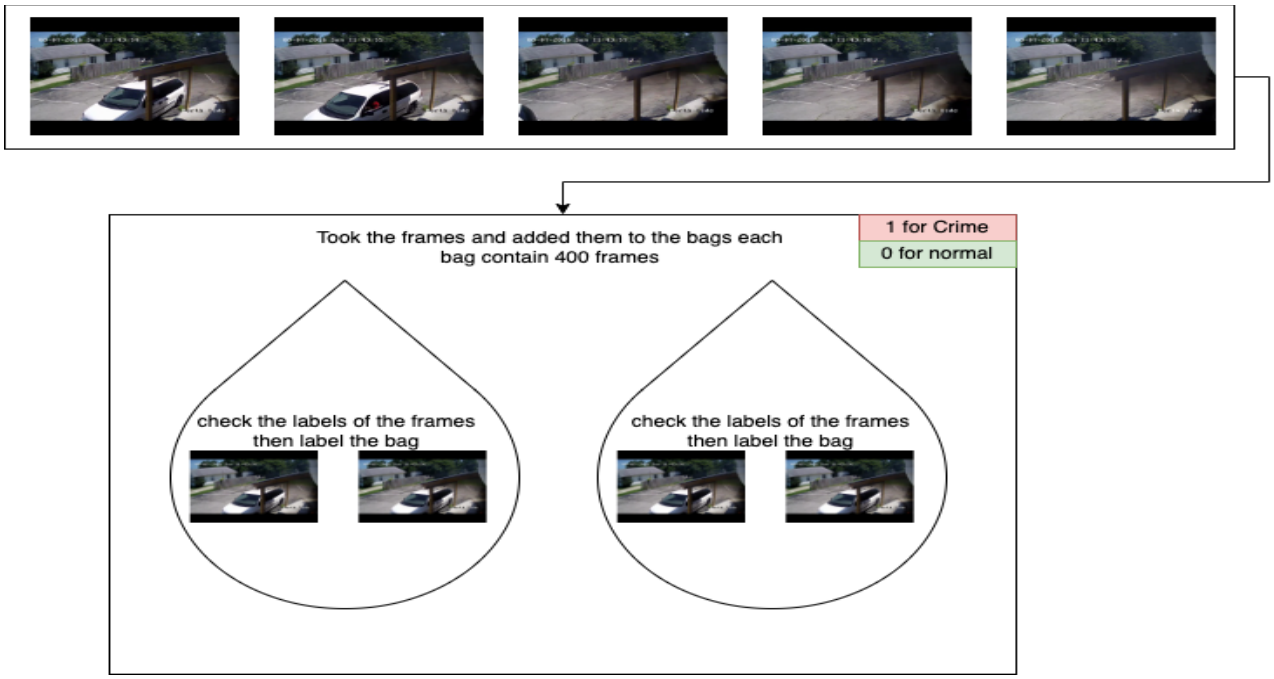

Figure 17 Adding Frames to bags

4.3.3 training process:

We will talk about the system's training procedure in this part. The bags are now kept in a numpy array with a shape of (400, 64, 64, 3), with an X size of (number of bags, 400, 64, 64, 3). Then, we begin training the 3D convolution neural network model, which accepts 5 dimensions of form rather than just 4, and all of the layers in the model are 3D with the exception of the long short term memory (LSTM), which is a 2D layer, as illustrated in (figure 18). We employ the SGD optimizer, which is an improved version of Adam since SGD generalises more effectively than Adam optimizer, throughout the training phase. We also establish our schedule for learning. Then, during training, we determine the training accuracy and the validation accuracy. Specifically, the model trains on 60% of the data, or the training set, giving us the training accuracy, and validates on 20% of the data, or the validation set, giving us the validation accuracy.

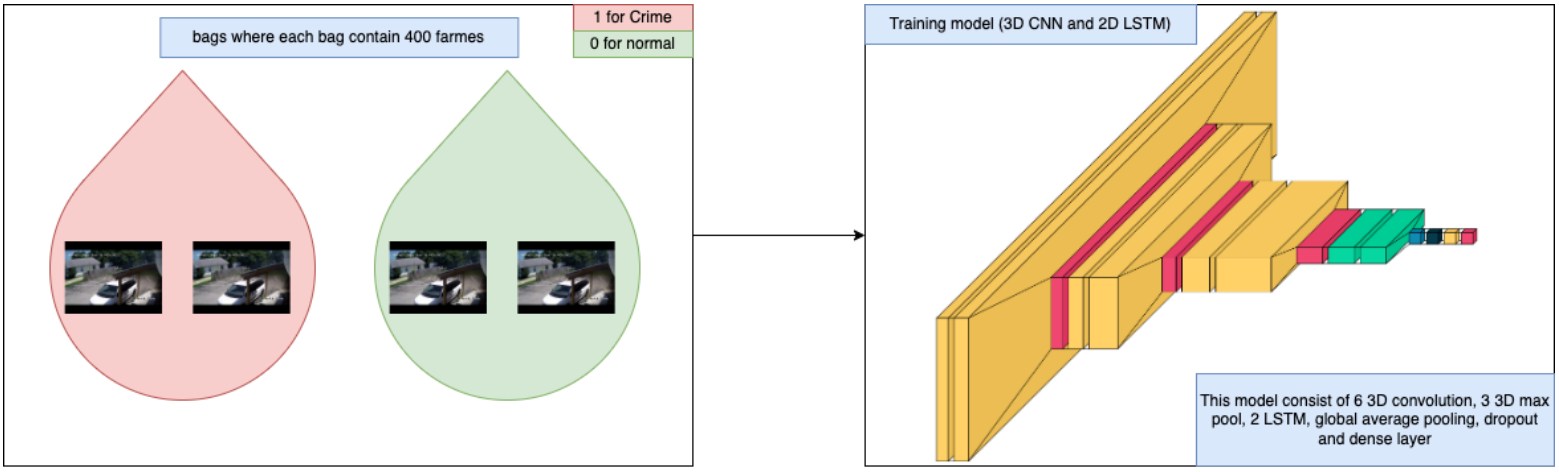

Figure 18 Training process

4.3.4 Testing process:

The model is prepared for testing after being constructed and trained. We first load the testing set, which comprises 20% of the datasets. We must also resize the frames, add them to the bags, and label them. However, unlike the training procedure, we must preserve these labels for later use and not provide them to the model. After doing this, we preprocessed the data on the test set in the same way that we had on the training set and the validation set. Additionally, submit the test to the model so that it may begin to forecast if each bag will be regarded as a crime or a typical circumstance. If the model gave us a 1, the bag is thought to be illegal; if it gave us a 0, the bag is seen to be normal, as shown in (figure 19). After the model has completed its predictions for all of the bags, we begin to construct our classification report, compute its characteristics, and determine the testing accuracy by contrasting the model's predictions with the labels we produced while making the bags. The generation and analysis of the confusion matrix will come as the final phase.

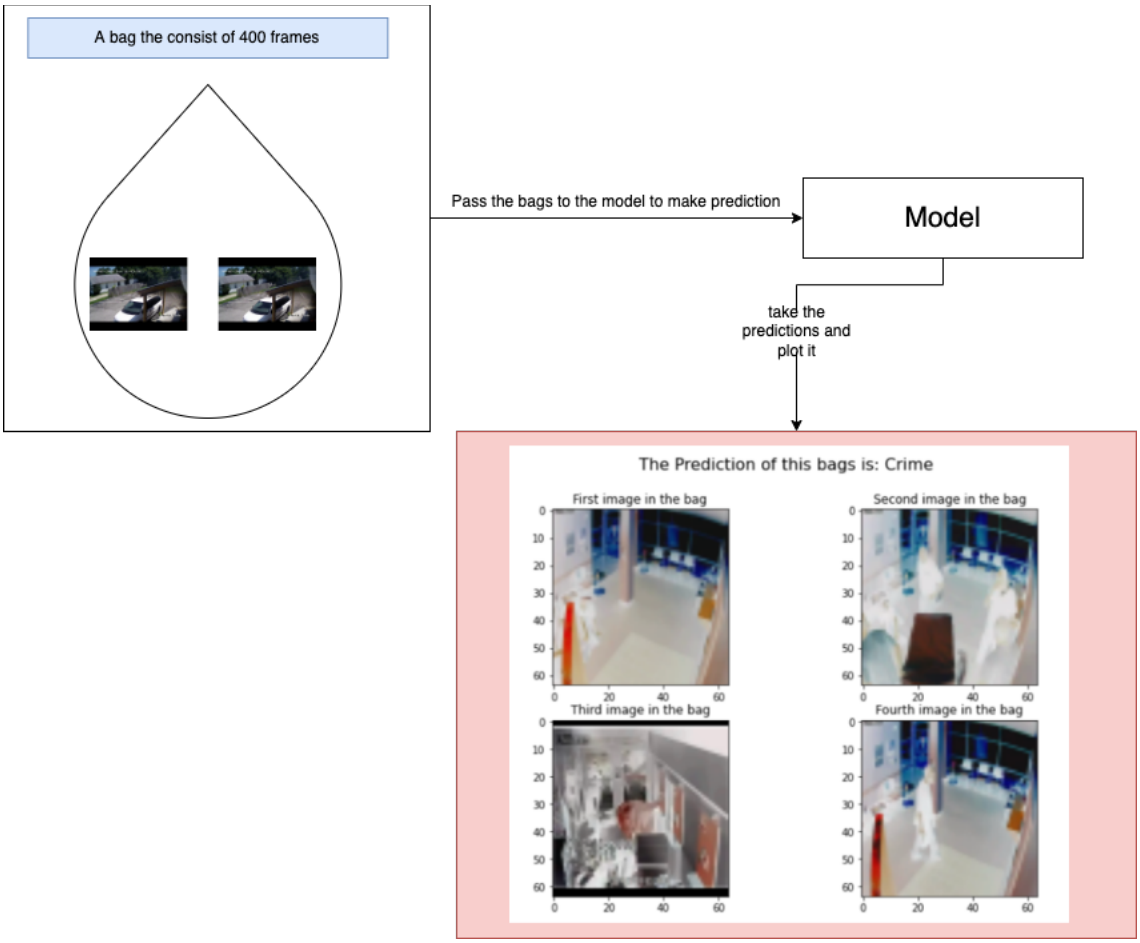

Figure 19 Testing Process

Chapter 5: Results and Evaluation

In the upcoming chapter the models performance that were made for this system will be compared together to show the model with the best metrics. This will be explained briefly by stating the testing methodology used and showing the results of each model. The limitations of this system will be explained briefly in this chapter also. Lastly the time performance of the model will be discussed stating exactly how much time it took for the model to learn.

5.1 Testing Methodology:

There are a variety of methods for evaluating how effectively a model predicts particular classes from a given dataset. In this section the testing methodology used will be illustrated in detail this section. First of all, the data was split into three thing 60% training, 20% validation and 20% testing. We were able to split the data in this way as the size of the dataset is large as it was explained previously. The validation data was tested at each epoch by passing the validation data and their labels. While training the model calculate at each epoch the training accuracy, training loss, validation loss and the validation accuracy. After some changes done to the model, reached the best metrics, we passed the testing data to the model to check if the model is predicting correctly or not. Lastly, calculated the testing accuracy and plotted the confusion matrix. To sum up the goal of this methodology is to get the least training loss, validation loss, training accuracy, validation accuracy and make sure that the model is predicting correctly by testing it from the test data.

5.2 Results:

5.2.1 Worst Case:

The worst case of this model is when the model itself had bad training accuracy. So, when the test data is passed to the model it made a wrong detections.

5.2.2 acceptable case:

The acceptable case of the model is that the model predicts some of the bags correctly and other wrongly but most of the bags are predicted wrongly.

5.2.3 Best case:

The best case of the model is predicting most of the testing bags correctly and have a very low validation and training loss and have a good training and validation accuracy. The best model consists of 6 convolution layers with activation RelU and 2 Long Short Term Memory (LSTM) layers with activation TanH. The convolution have kernel size of (3x3x3) and strides of (1x1x1) and the LSTM layers have kernel size (3x3) and strides (1x1). This model was trained for 90 epochs with batch size of 10. The following figure contain prediction of two bag where each bag contained 400 frames and the model predicted both correctly as shown in (figure 20).

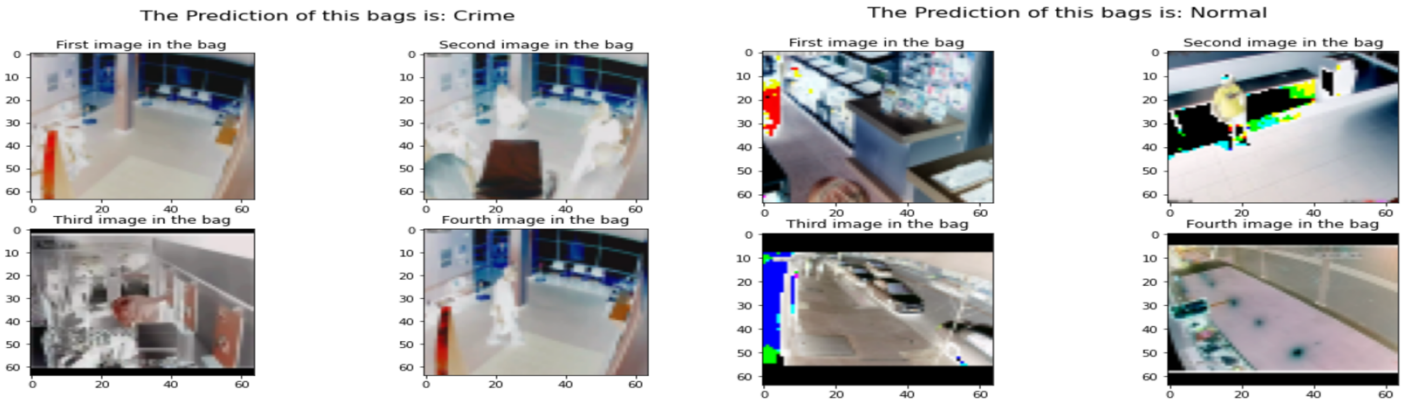

Figure 20 Output of the best Case

5.2.4 Limitations:

The biggest limitation in the project was learning that there is crime happening in this video taking into consideration the more than one frame. This is where the Multiple Instance Learning (MIL) idea came from. Where the model is taking into consideration 400 frame per instance in order to predict. Because of the size of the dataset that is very large as it consists of 370,000 frames from the crime videos and 360,000 frames for the normal videos the processing time of the model was not the very fast. Also, the model consisted of many convolutions layer and other 2 LSTM layers also those slowed down the model training. These frames when they were added to the bags come with 1,825 bags for training and validation and each bag contained 400 frames. This made the model take about 44 minutes per epoch and this was the best time per epoch to get.

5.2 Evaluation:

5.3.1 Accuracy Evaluation

The following table explains each model that was built for this system and their accuracies. Most of the models had been overfitted this made them to be biased to only 1 class of the data. Until used the LSTM which improved everything really quickly as shown in (table 1).

Model number	Number of convolution layers	Number of LSTM layers	Kernel size	strides	padding	Batch size	Training accuracy	Validation accuracy	Testing accuracy
Model 1	4 layers	0 layers	3x3x3	1x1x1	Valid	20	49%	66%	50%
Model 2	4 layers	0 layers	3x3x3	1x1x1	Valid	10	99%	100%	50%
Model 3	6 layers	0 layers	6x3x3	1x1x1	valid	10	92%	100%	50%
Model 4	6 layers	0 layers	3x3x3	1x1x1	valid	10	70%	50%	70%
Model 5	6 layers	2 layers	3x3x3	1x1x1	valid	10	87%	90%	95%

The fifth model gave the best results, and it was the best one of them in the prediction from the test data. In figure 2 show the classification report of the model that was showing the testing accuracy and support for each class. This model was really good with the normal class, and it was never predicted wrongly. And it was shown more with the confusion matrix of the model as shown in (figure 21).

	precision	recall	f1-score	support
0	0.92	1.00	0.96	161
1	1.00	0.88	0.94	116
accuracy			0.95	277
macro avg	0.96	0.94	0.95	277
weighted avg	0.95	0.95	0.95	277

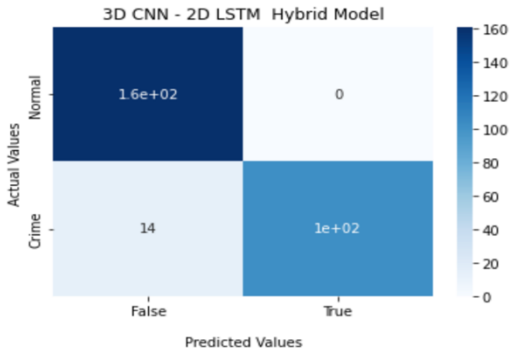

Figure 21 Report and Confusion Matrix

5.3.2 Model Time Performance:

Regarding the time performance of the model, it was one of the limitations of this system as the model had a lot of parameters and the data is very large that consisted of 730,000 frames. Google Colab had some problems so decided to work on the local computer. The model was trained on the CPU of the computer as MacBook M1 had the CPU and GPU integrated together which made it able to train such a model. Each epoch took about 2325 second and 2 second per sample. so each epoch took about 40 minutes for training only without validating. Which is 60 hours to complete the train of this model.

Chapter 6: Conclusion and Future Work

6.1 Conclusion:

In conclusion, the system that was developed in this project is a crime detection system which had a goal which is to detect if there was a crime in this video or not. There were different approaches used in this project. The 2D convolution was an option but the project needed to have a relation between the frames so, the 3D convolution was a better option to be used. Moreover, in order to use the 3D convolution, the Multiple Instance Learning approach was needed to be used where we added the frames to the bags and labeled each bag as explained in the previous chapters. Furthermore, due to the results got from the 3D convolution alone which was not good enough, a Long Short Term Memory layers were added to the model which made a huge boost to the results. The system architecture which had the LSTM layers gave the best result where the testing accuracy was 95%. Lastly the system is able to classify between crime events and the normal events with 95% testing accuracy.

6.2 Problem Issues:

6.2.1 Technical issues:

Due to the large data that was used in this system, kernel dying was a huge issue that was faced. This issue was overcomed by deleting any numpy array that is not used anymore which made debugging harder as the old every variable or numpy array is deleted after finishing the task that it was needed for.

6.2.2 Scientific issues:

Using both Convolution Neural Network and Recurrent Neural Network “CNN and RNN” was a challenge. This was because the difference between the shapes of the tensor that is accepted by the 3D CNN and the LSTM. Moreover, this step was needed the expected result from the 3D CNN was not met. This challenge was solved by using the 2D LSTM.

6.3 Future Work:

In the future work, classification between the 12 types of the crime is a goal so, the crime detection system can give an awesome result. A huge data from each type needed in order to each type in order to achieve this. If the multiclass classification is achieved with a great accuracy then the system will be fully automated no need for any human intervention.

References

1. Dorogyy, Y., Kolisnichenko, V., & Levchenko, K. (2018, September). Violent crime detection system. In *2018 IEEE 13th international scientific and technical conference on computer sciences and information technologies (CSIT)* (Vol. 1, pp. 352-355). IEEE.

2. Samuel, D. J., & Cuzzolin, F. (2021). SVD-GAN for Real-Time Unsupervised Video Anomaly Detection.

3. S. Chackravarthy, S. Schmitt and L. Yang, "Intelligent Crime Anomaly Detection in Smart Cities Using Deep Learning," 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC), 2018, pp. 399-404, doi: 10.1109/CIC.2018.00060.

4. *Unsolved Homicides*. (n.d.). Kansas City Missouri Police Department. Retrieved November 10, 2021, from https://www.kcpd.org/crime/unsolved-homicides/

5. Jonas DE, Wilkins TM, Bangdiwala S, et al. Findings of Bayesian Mixed Treatment Comparison Meta-Analyses: Comparison and Exploration Using Real-World Trial Data and Simulation [Internet]. Rockville (MD): Agency for Healthcare Research and Quality (US); 2013 Feb. Table 20, Advantages and disadvantages of the Bayesian MTC approach. Available from: https://www.ncbi.nlm.nih.gov/books/NBK126112/table/discussion.t1/

6. Jain, R., Nayyar, A., & Bachhety, S. (2020). Factex: a practical approach to crime detection. In *Data Management, Analytics and Innovation* (pp. 503-516). Springer, Singapore.

7. Council, G. (2018). Using OCR: How Accurate is Your Data? | Transforming Data with Intelligence. Retrieved 25 February 2022, from https://tdwi.org/articles/2018/03/05/diq-all- how-accurate-is-your-data.aspx#:~:text=Leveraging%20Your%20Document%20Data&text=Obviously%2C%20the%20accuracy%20of%20the,level%20of%20accuracy%20is%20acceptable.

8. U. V. Navalgund and P. K., "Crime Intention Detection System Using Deep Learning," 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET), 2018, pp. 1-6, doi: 10.1109/ICCSDET.2018.8821168.

9. Gupta, A. (2022). Difference between ANN, CNN and RNN - GeeksforGeeks. Retrieved 25 February 2022, from https://www.geeksforgeeks.org/difference-between-ann-cnn-and-rnn/.

10. Advantages and Disadvantages of TensorFlow. (2022). Retrieved 26 February 2022, from https://techvidvan.com/tutorials/pros-and-cons-of-tensorflow/

11. Sung, CS., Park, J.Y. Design of an intelligent video surveillance system for crime prevention: applying deep learning technology. *Multimed Tools Appl* (2021). https://doi.org/10.1007/s11042-021-10809-z.

12. Advantages of Deep Learning | disadvantages of Deep Learning. (2022). Retrieved 26 February 2022, from https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-Deep-Learning.html.

13. K-Nearest Neighbor(KNN) Algorithm for Machine Learning - Javatpoint. (2022). Retrieved 26 February 2022, from https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning.