# Project Title:- Crime Detection and Forensic using Deep learning

Team Members:-

Mahendran.K (112720104020)
Siva.R (112720104036)
Chandrakumar.M (112720104301)

# St.Peter's College of Engineering and Technology

This project is done by the Department of Computer Science and Engineering and done by the Team Members R.Siva (112720104036, M.Chandrakumar (112720104301) and Mahendran.K (112720104020) of St Peter's College of Engineering and Technology, Avadi, Chennai -600054.

# 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 countries do not have the manpower to monitor the all the events and places, so authorities can implement an intelligent crime detection system using deep learning. Suggesting utilizing both regular and abnormal videos to learn anomalies. 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 action that is made by a person who is considered as a criminal and this action can cause a physical harm to another person or damage or loss of a property. The person doing this action is always punished by the authorities of the country according to the severity of the crime. Every culture has a significant amount of crime. Its costs and consequences affect almost everyone to some extent. Costs and impacts come in a broad variety of shapes and sizes. Furthermore, some costs are temporary, while others endure a lifetime. Of course, the ultimate price is death. Medical expenses, property losses, and lost wages are some of the other expenditures that victims may incur. The discovery of a crime, the identification of a suspect, and the gathering of adequate evidence to indict the person before a court are the three distinct aspects of crime detection as shown in (fig.1). If a Crime detection systems applied on the cameras on the streets police would solved a crime happened on January 10, 2017 the victim name is Donald Coty Jr the victim was found in the driver seat he has been shot and passed away on the spot, this crime happen on 12 and Paseo [4], if the crime detection system applied on any camera in the location of this crime authorities would be able to save this person or capture the criminal. The goal is to help capturing the criminal and prevent his escape to be able to provide the society with a safer life also to alert the police officers that there is a crime happening now so authorities can be able to fasten the process and not waiting for someone to report the crime to the police to make it easier to capture the criminal. The current situation method of crime detection is to detect the criminal by using different computer vision algorithms, Bayesian methods, artificial neural networks, spiking neural networks and fuzzy logical approach [1]. The Bayesian methods has some pros such as:

1. Ability to combine direct (from head-to-head trials) and indirect (from placebo-controlled trials) findings into a single analysis.
2. Capable of producing results for all relevant comparisons inside a linked network.

And has some cons such as:

1. Many investigators find the analyses difficult to understand since they are frequently conducted using software that is foreign or unknown to them (usually run using WinBUGS).
2. It necessitates a higher level of statistical knowledge than some other approaches [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 and to be able to adapt according to current data. Also, to track all places and 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).

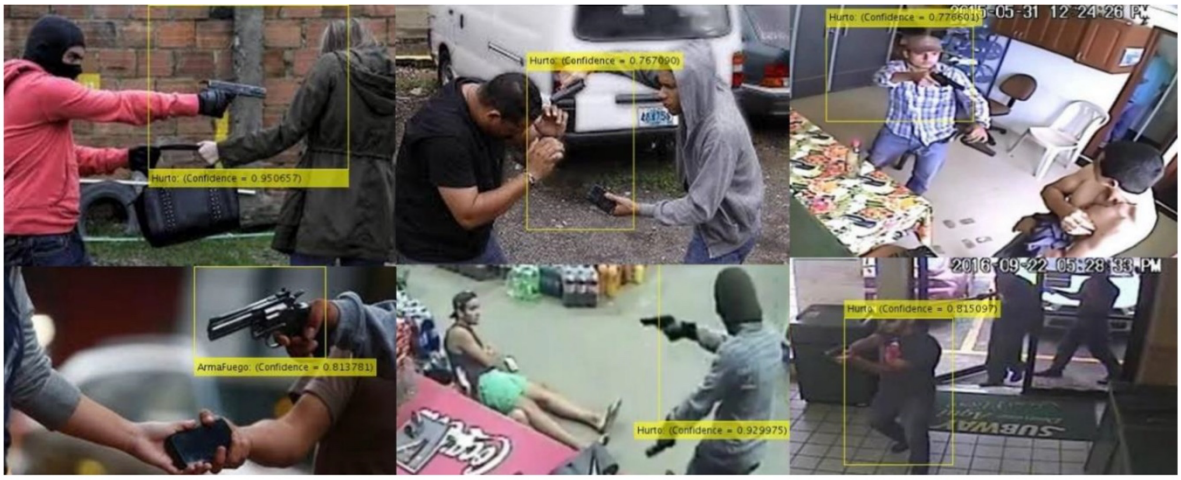


*Figure 2 Weapon Detection*

### 2.1.1 Previous algorithms

**Deep Learning:** Machine learning has a subclass called deep learning. Deep learning is a type of machine learning that learns characteristics and tasks from data. Images, text files, and sound files are all examples of data [12].

**K-Nearest Neighbors algorithm (KNN):** The KNN method is based on the Supervised Learning approach and is one of the most basic Machine Learning algorithms. The KNN method assumes that the new case/data and existing cases are comparable and

places the new case in the category that is most like the existing categories. The KNN method saves all available data and classifies a new data point based on its similarity to the existing data. This implies that fresh data may be quickly sorted into a well-defined category using the KNN method [13].

## 2.2 Previous Work

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

#### 2.2.1.1 Strategy and structure:

Road crime is a big issue that all modern cities are dealing with nowadays. Many criminals use road transportation as a means of emigrating. Due to a lack of proof, thefts and many other crimes go unreported and unsolved. The main purpose of this research is to detect the criminals while they are escaping using vehicles by running the license plates of the vehicles across a database or walking on the street by running their faces across criminal faces database as shown in (fig.2). The system used in this research is the OCR (Optical Character Recognition). K-Nearest Neighbors algorithm (KNN) is employed as a text recognition algorithm in the suggested system, and HAAR Faces Classifier and SVM are used to conduct face recognition. [6]

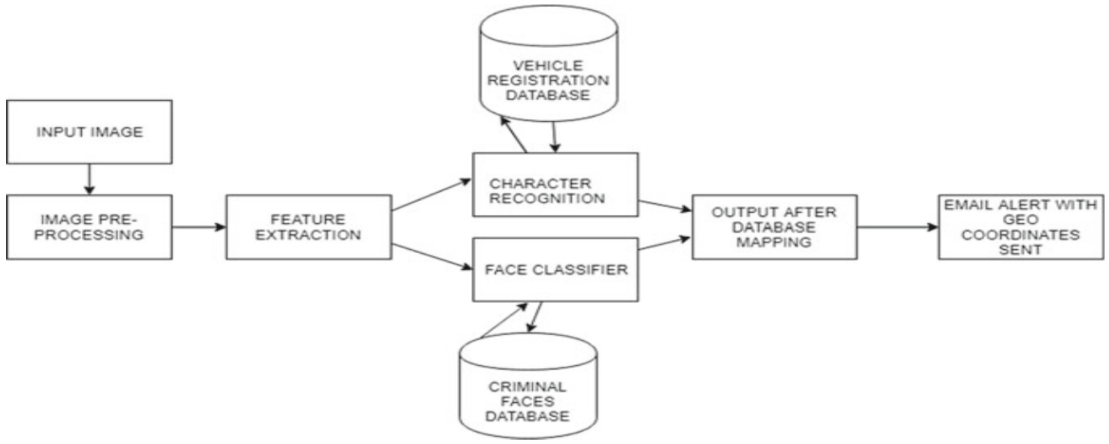


*Figure 3 System Architecture [6]*

#### 2.2.1.2 Data

The researchers are the ones who created the dataset, but it has an issue that it only consists of 50 images where each image show facial expression [6].
The dataset is made up of photos captured by a camera of a single person. It's a limited dataset but still it can be viewed as a milestone for better monitoring.

#### 2.2.1.3 Method evaluation

The OCR system is very accurate it can read the license plates it can give 98% or 99% accuracy [8]. However, the OCR is still highly time consuming and lacks any facial identification for the detection of the crime [6]. KNN algorithm is one if the simplest algorithm and gives a high accuracy. But KNN require a lot of memory and unrelated characteristics and noise have an impact on output.

#### 2.2.1.4 Results evaluation

The mechanism for detecting crime has been upgraded, with far more accurate results and increased efficiency. It makes use of both text and facial recognition technologies. The KNN method uses a Gaussian Blur filter to cope with all sorts of photos for number plate identification, regardless of noise, intensity, or other characteristics. Similarly, the self-built database predicts far more accurate outcomes in terms of lighting conditions and human movement for face recognition. In ambient light circumstances, the suggested system works with an accuracy of more than 85% successful detections.

### 2.2.2 Research 2: Crime Intention Detection System Using Deep Learning

#### 2.2.2.1 Strategy and structure

In this research they discussed that the CCTV are widely used all over the world, but these cameras don't do anything to control the crimes itself those cameras just monitor the crimes so that the police officers can use them when someone contact the police about the crime. They used pre-trained deep learning models. The project's major goal is to identify weapons in less time with more accurate findings and fewer false positives than machine learning approaches, as well as to make CNN work without performance degradation with less training data. Pre-trained models, such as GoogleNet and VGGNet-19, have been trained on millions of photos and can recognize objects in

fresh images with minimal mistakes as shown in (fig.3). They chose the VGGNet19 model because of its excellent training accuracy. It properly classifies and recognizes items. Input frames are accepted through the input layer, which performs pre-processing. Preprocessed images are then passed to the Convolution, Max-pooling, and FC layers, which perform feature extraction, filtering, mapping, and classification. Finally, the output layer sends a crime intentions security message using a registered API if any crime intentions are detected [8].

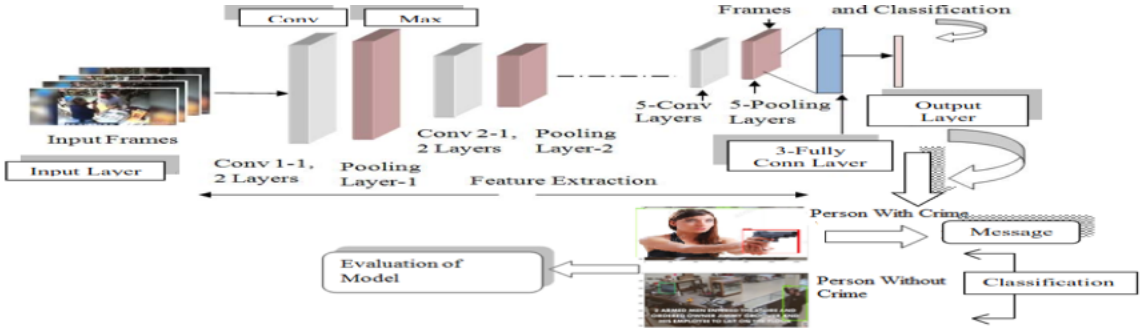


*Figure 4 System Architecture[8]*

#### 2.2.2.2 Data

The designed system has been tested with datasets for both videos and images that have been collected from YouTube and Google. The datasets collected are of the type of robbery, murder, and other illegal activities with weapons in hand, where the use of weapons is strictly prohibited in areas such as ATMs, banks, and other public places [9].

#### 2.2.2.3 Method evaluation

In this research, they used the Convolution Neural Networks (CNN) which is a deep learning algorithm, GoogleNet and VGGNet-19 which are some pre-trained models. In image identification challenges, CNN provides extremely high accuracy. Still, the location and orientation of an object are not encoded by CNN. In addition, a large amount of training data is necessary [9].

#### 2.2.2.4 Results evaluation

This project has been trained on two pre-trained models VGGNet-19 and GoogleNet and both gave different accuracy. VGGNet-19 gave the higher results with average accuracy 92%. Moreover, GoogleNet model gave average accuracy of 69%. Also VGGNet-19 needed less computational time than GoogleNet.

### 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

The goal of this research is to create an intelligent video surveillance system that can actively monitor in real time without the need for human intervention. After constructing an artificial intelligence server and a video surveillance camera, deep learning technology will be used to solve the difficulties of the existing video surveillance system through the data processing model design to display data for crime detection. In addition, this design presents an intelligent surveillance system that uses real-time processing to deliver a video picture and a notification message to the web to detect crimes rapidly and efficiently. In identifying crime and disasters, the suggested methodology demonstrates a good mix of speed and accuracy. This suggested model sends photos and notifications to a user's app in real time. Unlike previous suggested systems, this one allows users to instantly identify and detect threats using real-time visual data via socket connection. Video streaming is also possible while artificial intelligence deep learning is continually delivering real-time picture frames. Python flask for WEB, python TensorFlow for deep learning, and python socket for Raspberry Pi are all part of the system environment. The GPU server has four vCPUs with 30G RAM and one Tesla p40 GPU with 24G memory as shown in (fig.4). The job of raspberry pie is like that of the present video monitoring system. It takes photographs and serves as a transmission function for a camera image frame. Three functions are performed by the GPU server: socket communication with the Raspberry Pi, automated recognition deep learning and notification algorithm, and website opening. Within the GPU server, each

function is composed of three threads that run in parallel, thread configuration of the GPU server. Crimes can be prevented more proactively by giving real-time notifications to the user's application [11].

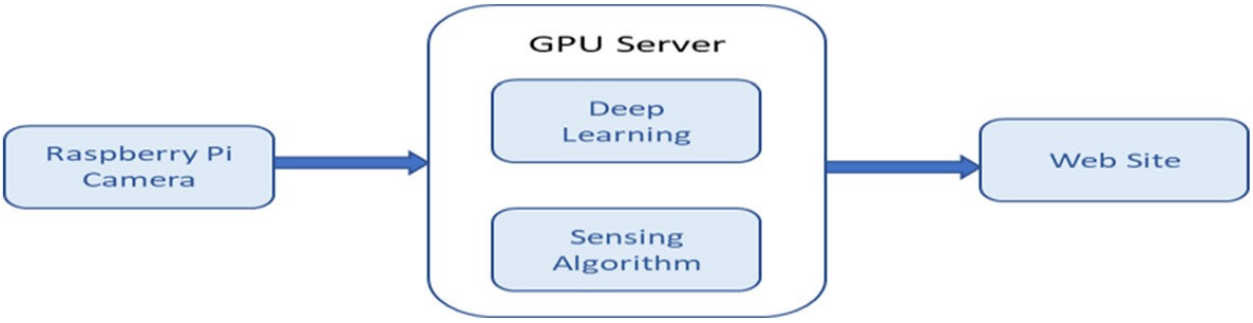


*Figure 5 System Architecture [11]*

#### 2.2.3.2 Data

The dataset used in this research from COCO net dataset, which is a large-scale dataset for object identification, segmentation, and captioning. Specific items, such as lethal weapons and flames, however, are not detected by the COCO net dataset. It must be directly taught for recognition by tagging more photos [11].

#### 2.2.3.3 Method evaluation.

The researcher is using python Tensorflow for deep learning which is a very scalable. Tensorflow can be used to accomplish almost any task. TensorFlow's ability to be put on any machine and provide a visual representation of a model enables users to create any type of system [10]. This system is not easy to implement due to the functions made by the system environment are not that easy [11]. Also, This system will not be the fastest as Tensorflow is not that fast.

#### 2.2.3.4 Results evaluation

This study presented an intelligent surveillance system that detects and guards against criminal activity. If deep learning technologies is used to the servers connected to the notification system, this suggested model proposes that criminal and catastrophe alerts may be made faster and more accurately [11].

# Chapter 3: Material and Methods

## 3.1 Materials

### 3.1.1 Data

In this project UCF-Crime Dataset will be used. 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

Anaconda software will be used, which is one of the most popular platforms, it's an open-source platform, it has more than 1500 python packages, it simplifies management and deployment of Artificial intelligence (AI) and machine learning. Python 3 will be used as it is easy to read, write and learn programming language and it's an open-source and free language. Keras is a high-level neural networks library built in Python, which makes it exceedingly straightforward and intuitive to use. Theano is a Python package that makes it possible to quickly assess mathematical operations, such as multi-dimensional arrays. It is mostly utilized in the development of 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

This section will discuss and clarify the solution process as well as the strategy that is 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. Multiple Instance Learning (MIL) will be used which is a type of the supervised learning. Furthermore, instead of receiving individually labeled instances the system will 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 train a 3D convolution neural network. A recurrent neural network will be also created. lastly, check the accuracy if the system reached an acceptable accuracy.

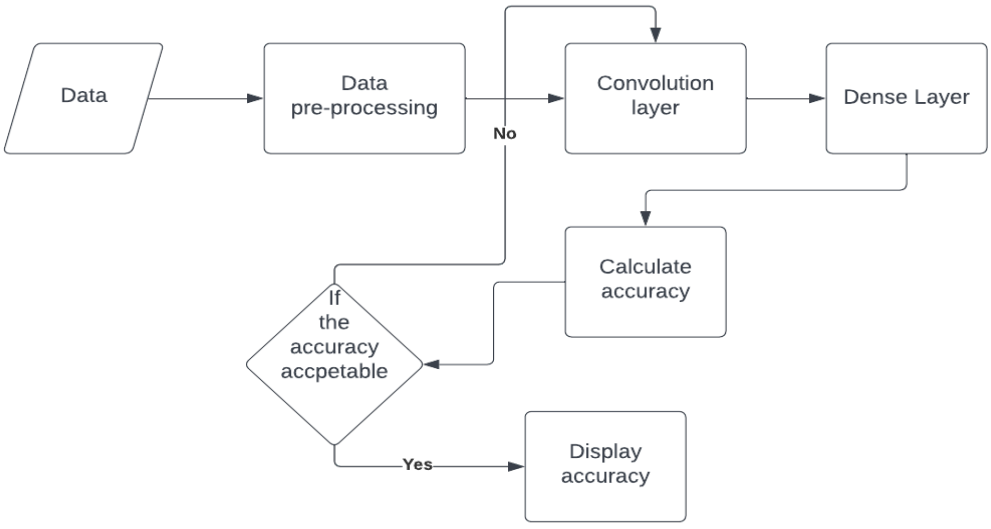


*Figure 7 proposed system Architecture*

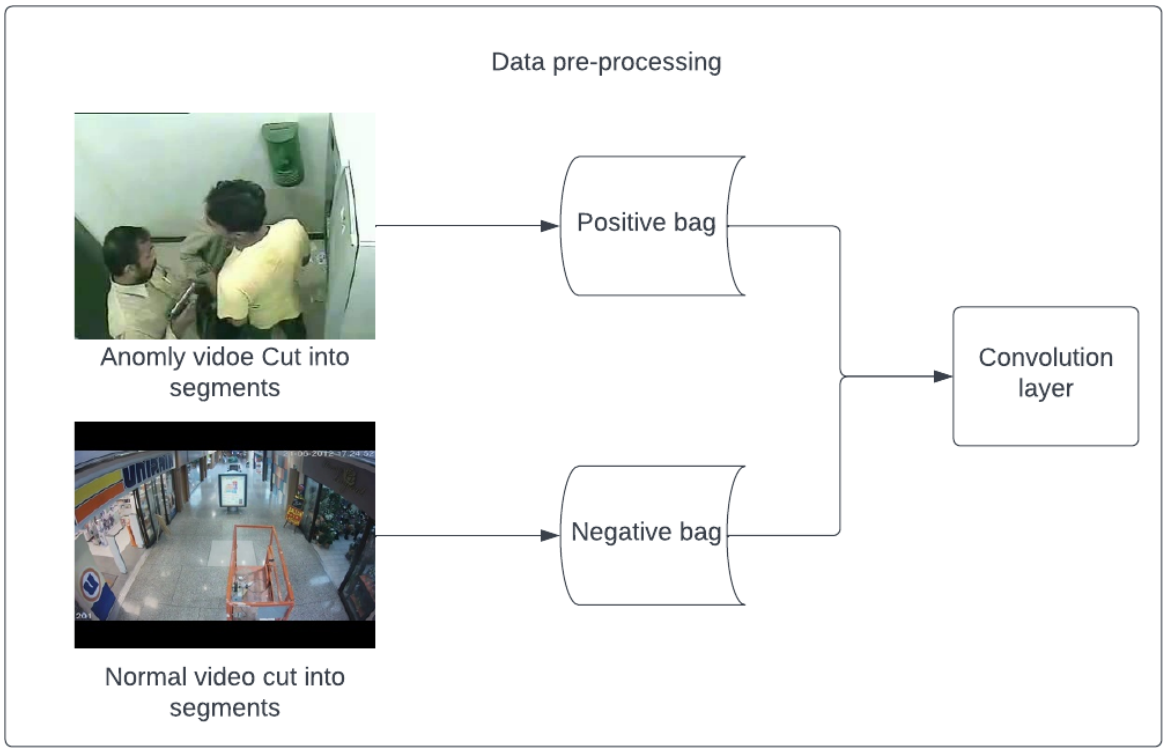


*Figure 8 Data preprocessing*

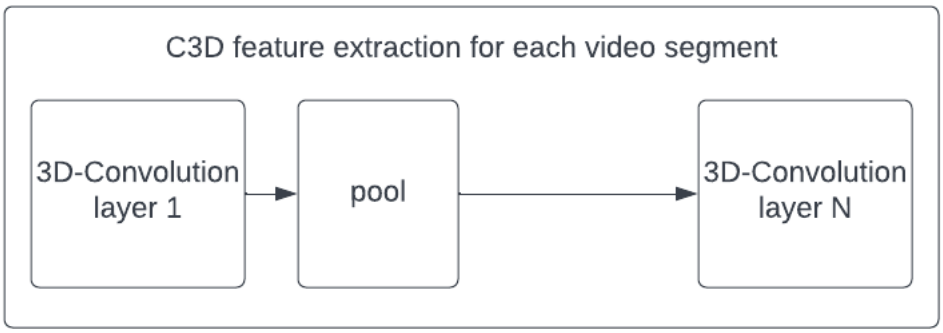


*Figure 9 3D Convolution Network*

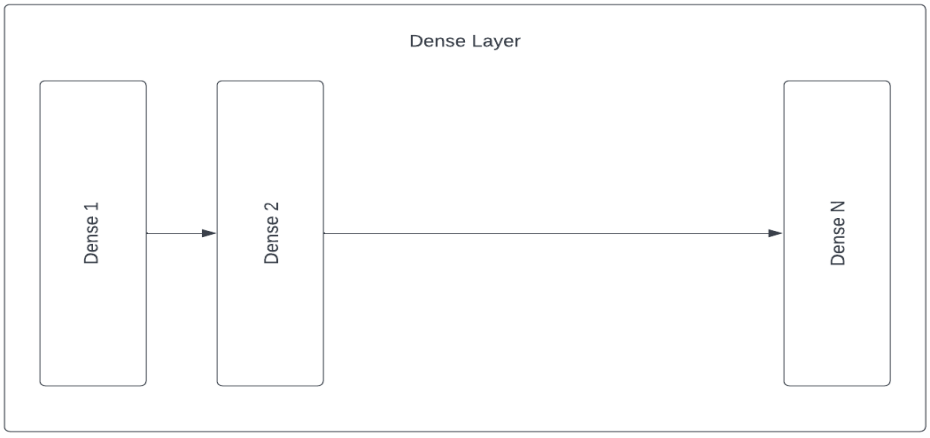


*figure 10 Dense Layer*

# Chapter 4: System Implementation

This section will go deeper into the technical aspects of the system's effective implementation. The technical components of the system, as well as the methods necessary to construct it, will be studied on the one hand, while the design decisions made in order to achieve the system's aim, will be thoroughly reviewed on the other. In the second half, how to put the system into practice will be explained. The architecture of the system, as well as the roles of each of its components, will be fully investigated in the second section. Finally, it will display the chapter's final section, which will display each system's component inputs, and also the outputs that correspond to them.

## 4.1 System Development:

The crime detection system was ultimately finished after a series of changes that went through several stages. The goal of this system is to detect if the video contains a crime or not. The dataset was allocated in the format of videos, so firstly the videos were cut into frames and then saved to the computer. A folder was created for each video were that folder contained all its frames. After cutting the videos to frames started to prepare the data for adding it to the bags (MIL algorithm). The frames had a size of 320x240 so, the frames were resized to become 64x64.

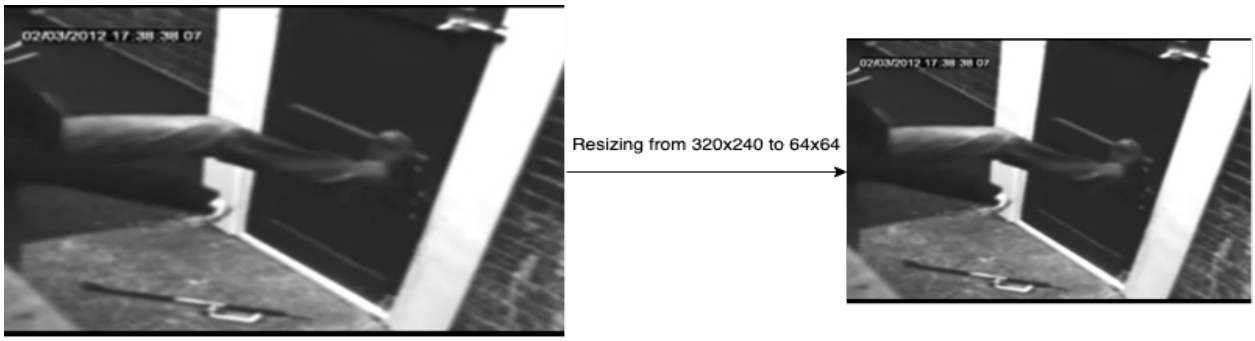


*Figure 11 Resizing Frames*

Then started to iterate on all the frames and take each frame and add it to a bag until the count of the frames inside the bag is 400 frame then check if there at least 1 frame that contain a crime then the whole bag will be labeled as crime if not the bag will be labeled as normal this is called the MIL (Multiple Instance Learning) and by adding these frames to bags then the MIL is applied. After this 3D CNN (3D Convolution neural network) was created using Keras, this 3D CNN model was made to check if it will achieve good results, or it will not. After testing the 3D

CNN, a 2D Long Short-Term Memory (LSTM) model is created, In the field of Deep Learning, LSTM is employed. It's a type of recurrent neural network (RNN) that can learn long-term dependencies and can help with sequence prediction. LSTM incorporates feedback connections, which implies it can grasp the entire sequence of data, in addition to single data points like photographs. The LSTM is a kind of RNN that performs exceptionally well on a wide range of issues. Then split the data into training, validation and testing where the training was 60% for the training set, 20% for the validation set and 20% for testing. Then combined both the 3D CNN and 2D LSTM layers are combined with a fully connected layer and then an output layer. During the development of this system several software were used such as OpenCV, Tensorflow, Keras, Numpy, Matplotlib, Google Colab, Jupyter Notebooks, Pycharm. Each bag was labeled, each bag contained 400 frames. The bags were labeled 1 for the bags that contained crimes and 0 for the normal bags.

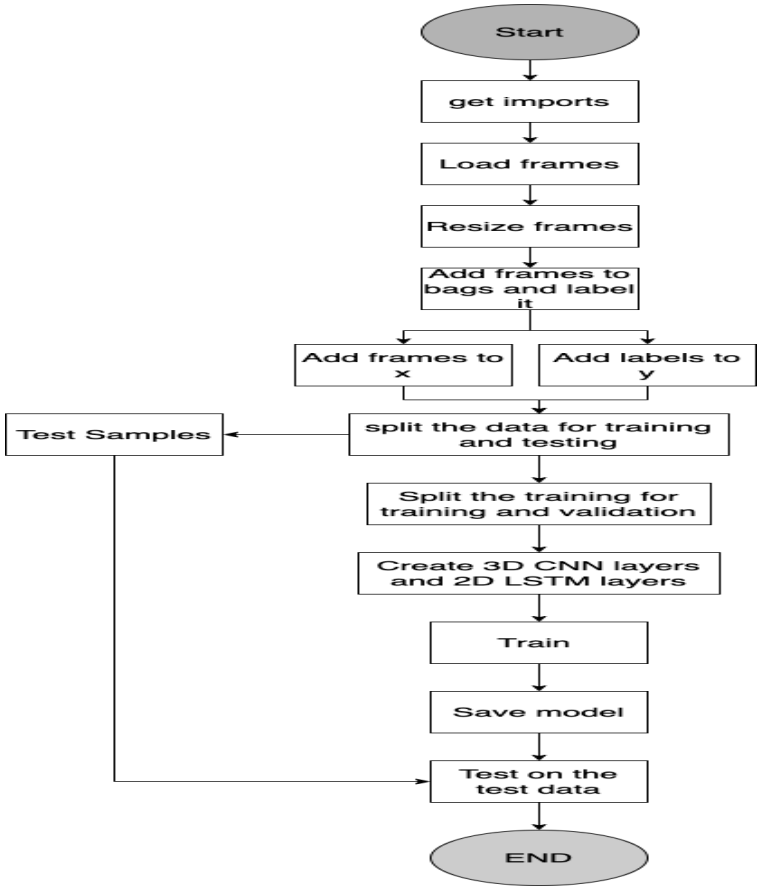


*Figure 12 Diagram for system development*

## 4.2 System Structure:

This information is provided in the first part's components. The whole system is already well defined, and the first part of this article discusses how data is transferred between the multiple elements. Will illustrate a tensor-board graph and systems that match these design requirements, as well as their functionality, in the second clause of the chapter.

### 4.2.1 System overview:

There three stages that were combined to give the crime detection system. The first stage has several steps. The first step is reading of the videos, second step is cutting each video in to frames and label each frame, third step is resizing each frame form 320x240 to 64x64, fourth step is adding frames to the bags and then label each bag where each bag contains different 400 frames. Labeling of bags were made by checking each frame inside the bag if there was at least 1 frame that was labeled as a crime then the whole bag is labeled as crime if the 400 frames did not contain any frame that contained a crime, then the whole bag is labeled as normal and the last step of this stage is to split the data into 60% training, 20% validation and 20% testing. This stage is considered the data preprocessing stage. The next stage is the training stage where there was CNN model and LSTM model combined together. In this stage this the system trains the combined model on the bags that was made on the previous stage were the X will contain the bags and the Y will contain the labels and the last step in this stage is to calculate the training accuracy and the validation accuracy. Then will reach the final stage where the system tests the model using the test set that were have been made in the first stage, then generate the classification report that contain the precision, recall, f1-score and the support. Then

calculate the testing accuracy and lastly generate the confusion matrix. The whole system stages as shown in (figure 13).

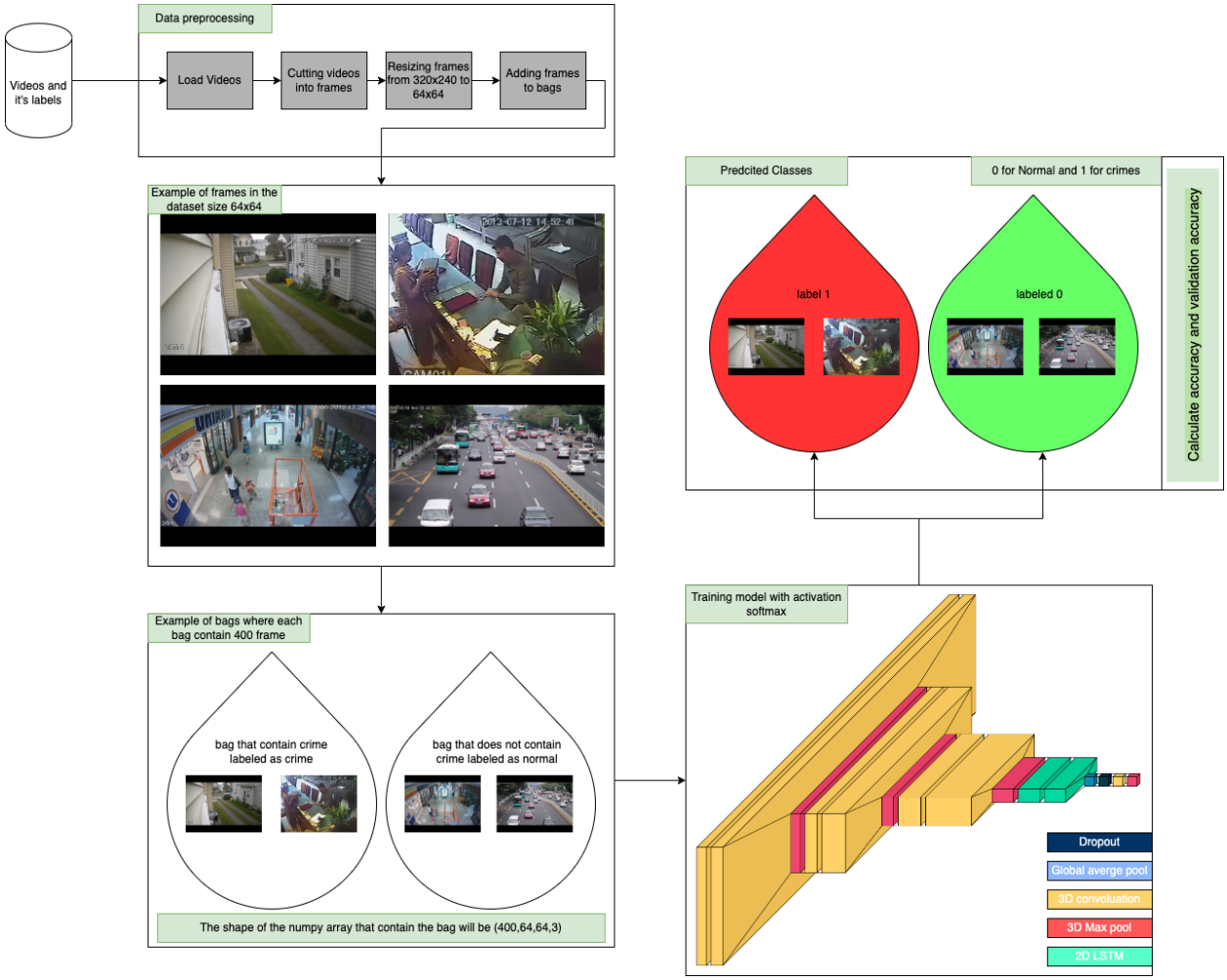


*Figure 13 System Overview*

### *4.2.1.1 CNN & LSTM Architecture Overview:*

The model was trained on the numerous iterations so the model can differentiate between the normal videos and the videos that contain crimes. Here is an illustrated diagram of the combination of the two models shown in (figure 14). The 3D CNN start with input shape (none, 400, 64, 64, 3) then 6 3D convolution layers with kernel size (3x3x3), strides (1x1x1), filters 2, 4, 8, 16, 32, 64 respectively and activation function RelU. After each two 3D convolution layer is separated with 3D max pool layer with pool size (2x2x2) and strided (2x2x2). Then after the third max pool two 2D LSTM layers with kernel size (3x3), 64 filters,

strides (1x1) and activation TanH. After this Global average pooling layer was added, dropout layer (0.5), Dense layer with 2 units lastly the output layer with activation Softmax.

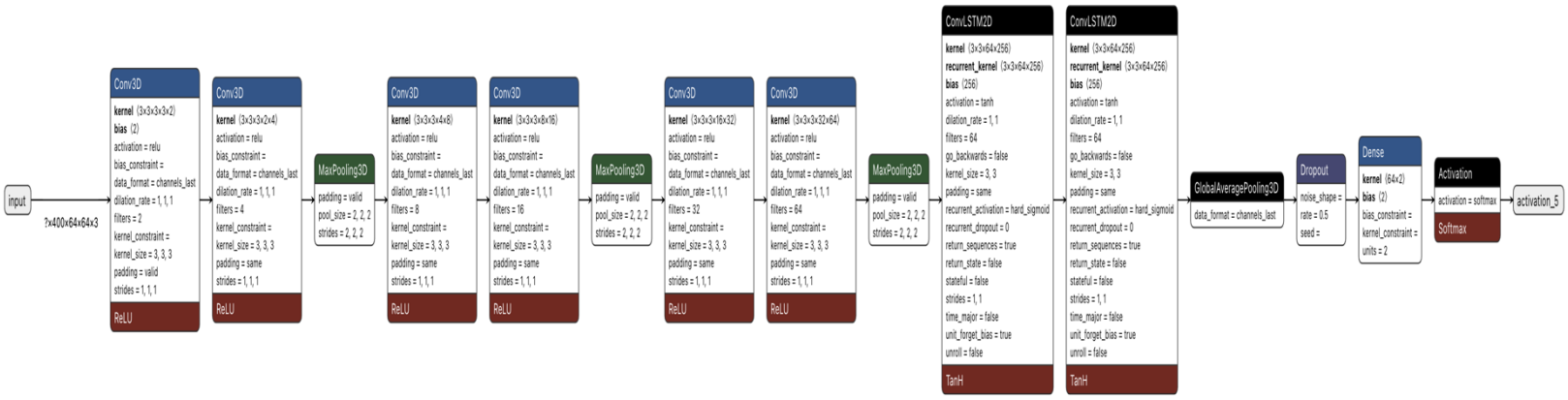


*Figure 14 Model Architecture*

### 4.2.2 TensorBoard:

The following diagram show the tensorboard diagram that show every thing the architecture of the system

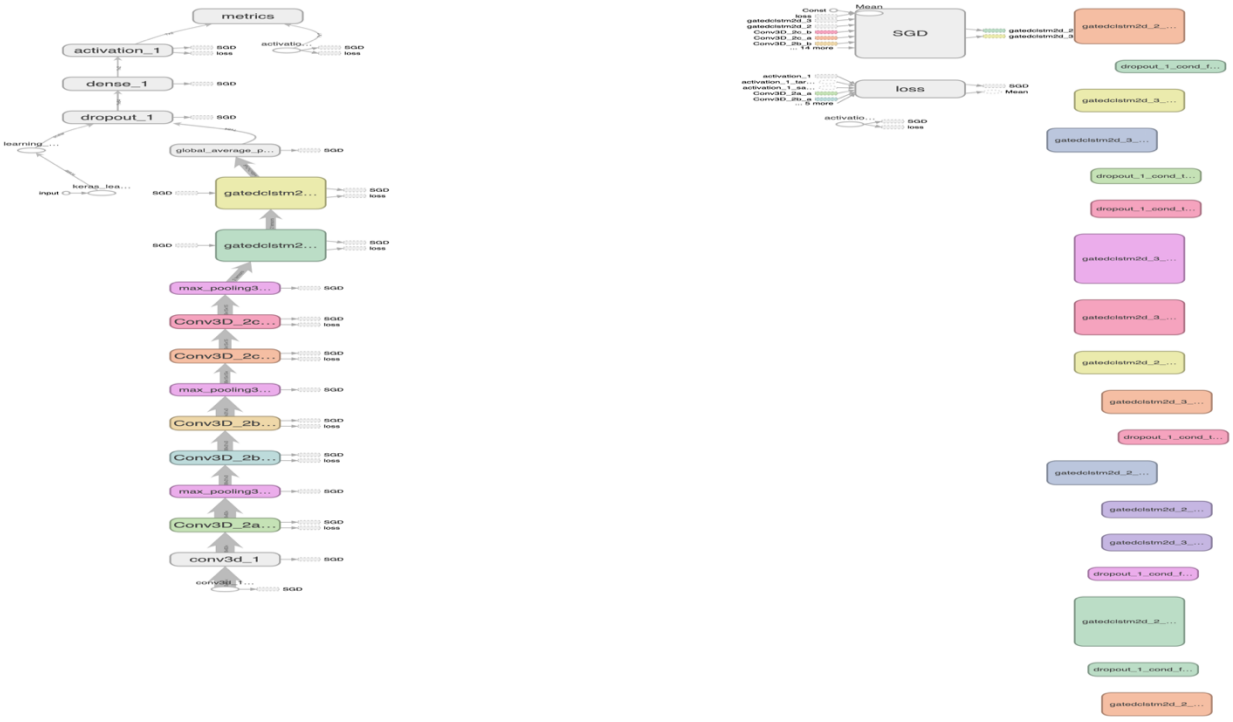


*Figure 15 Tensor Board*

## 4.3 System Running:

This section will discuss the input and output of the system, where each stage will be explained in more details.

### 4.3.1 Data Selection:

At first, the system read the videos and then cut these videos into frames. where 60 frames per second are took. where the shape the size of the image will be 320 x 240 which will be resized later as shown in (figure 16).


*Figure 16 Cutting Frames*

### 4.3.2 Data preprocessing:

The system read the frames that it has been cut in the previous section that will give us size of 320 x 240 that will be resize it to 64 x 64. Then create the bags by iterating over all the frames and take each 400 and add them to a bag. After adding the 400 frames to the bag then check if there is any frame that contain a crime then the whole bag will be labeled as crime. And if the 400 frames do not contain any frame that contain a crime, then the bag will be labeled as normal as shown in (figure 17). Then add bags to the X and the labels to Y. and then split the data in 60% training, 20% validation and 20% testing. And by this the data preprocessing section is finished. And move to the training process.

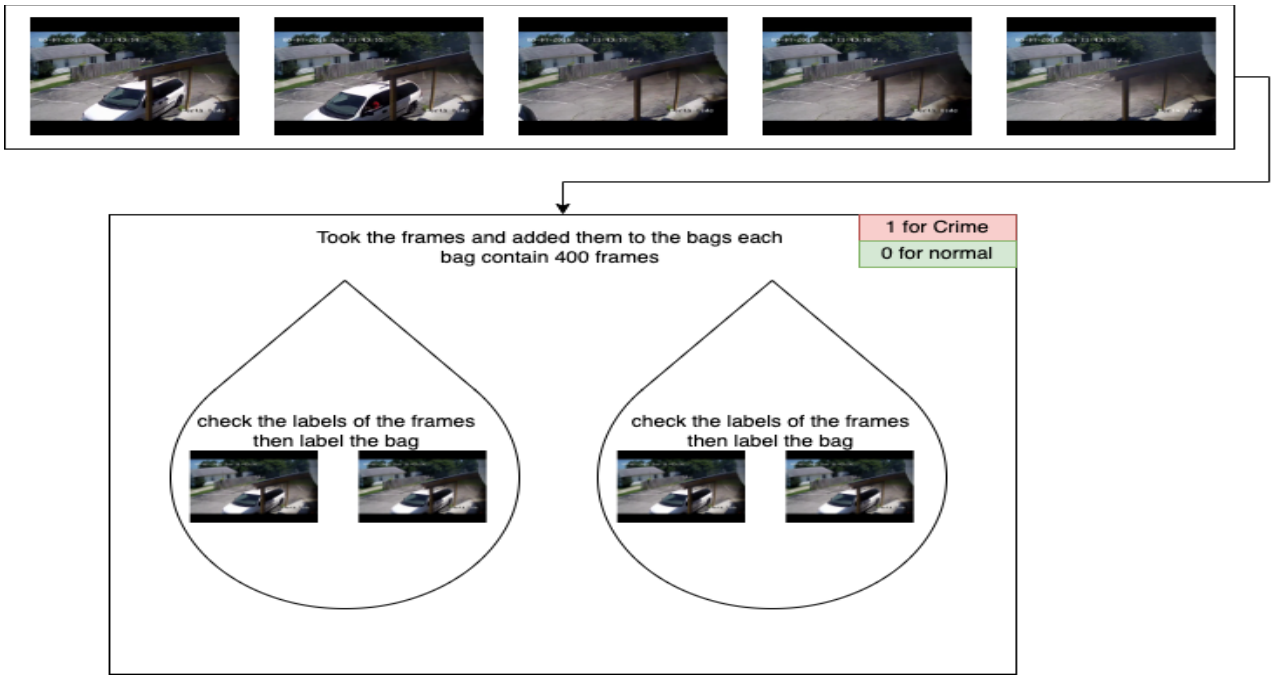


*Figure 17 Adding Frames to bags*

### 4.3.3 training process:

In this section how the training process is made on the system will be discussed. Currently the bags are stored in a numpy array where the shape of the bag is (400, 64, 64, 3) and the size of the X is (number of bags, 400, 64, 64, 3). Then the 3D convolution neural network model start training as the 3D convolution neural network accept 5 dimensions shape not as the 2D convolution neural network that accept 4 dimensions and every layer in the model is also 3D except the long short term memory (LSTM) is 2D layer as shown in (figure 18). During the training process SGD optimizer is used that is a better version of Adam as SGD generalize better than Adam optimizer. Then during the training, the model calculate the training accuracy, and the validation accuracy during each epoch. The model train on the 60% of the data which is the

training set this gives the training accuracy and validate on 20% which is the validation set which gives the validation accuracy.

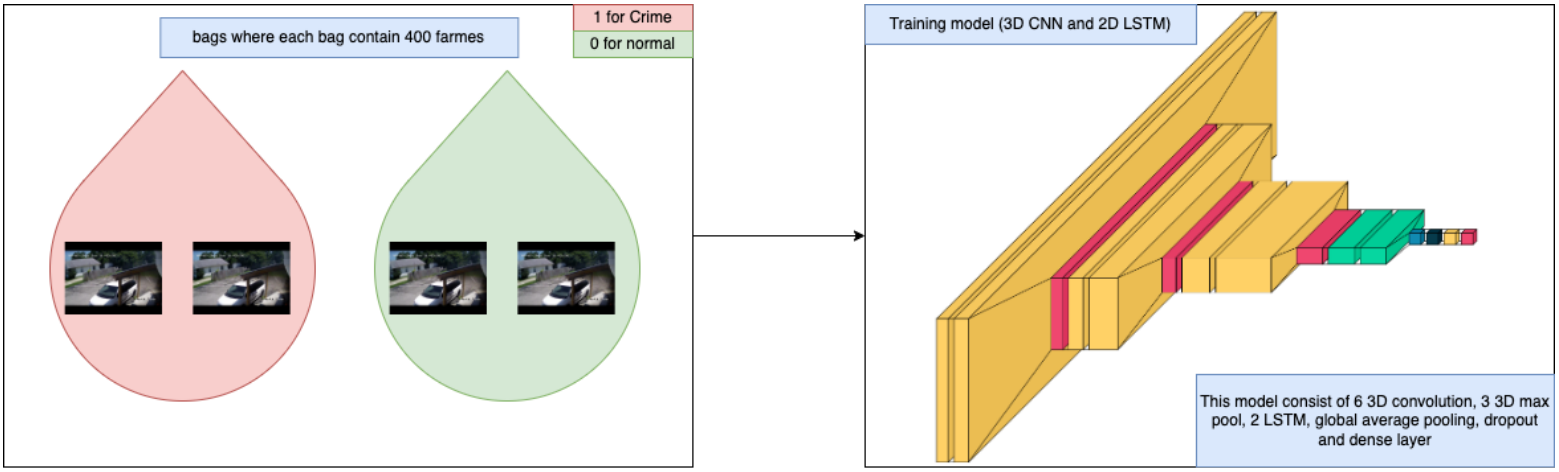


*Figure 18 Training process*

### 4.3.4 Testing process:

After building the model and training it then the model is ready for testing. First of all, the system loads the testing set which is 20% of the datasets. The test set must be resized also and to add them to the bags and label them also but save these labels for future use and don't pass it to model not like the training process. After performing this, created the data preprocessing on the test set as it was applied it on the training set and the validation set. Moreover, pass the test to the model so the model can start to predict whether each bag it considered as a crime or considered as a normal situation. If the model gave 1 then the bag is considered as a crime, if the model gave us 0 then the bag considered as normal as show in (figure 19). After the model finish prediction on all the bags, start to build the classification report and calculate its attributes. Then get the testing accuracy by comparing the results that the model gave us to the labels that we generated while creating the bags. And the last step will be generating the confusion matrix and analyze it.

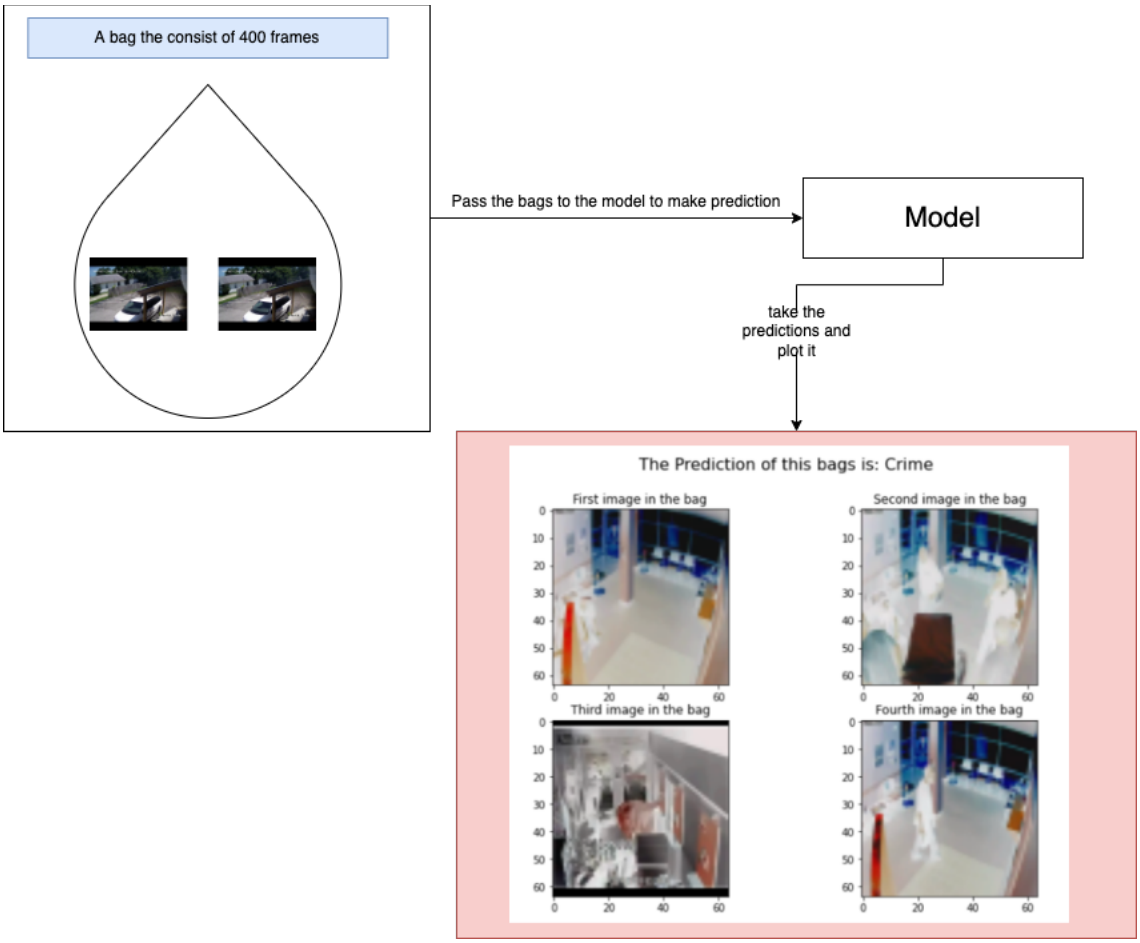


*Figure 19 Testing Process*

# Chapter 5: Results and Evaluation

In the upcoming chapter the model's 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. 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, 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 and validation loss and highest training accuracy and 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 detection.

### 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 has 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).

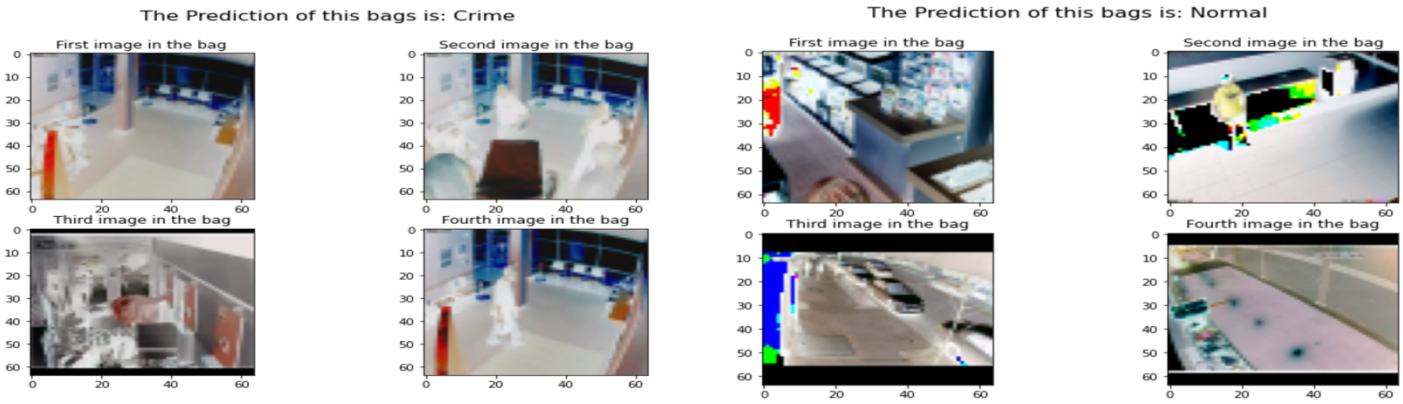


*Figure 20 Output of the best Case*

### 5.2.4 Limitations:

The biggest limitation in the project was learning that there is crime happening in this video taking into consideration 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 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 21) 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 did not predicted wrongly. And it was shown more with the confusion matrix of the model

| | 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 |

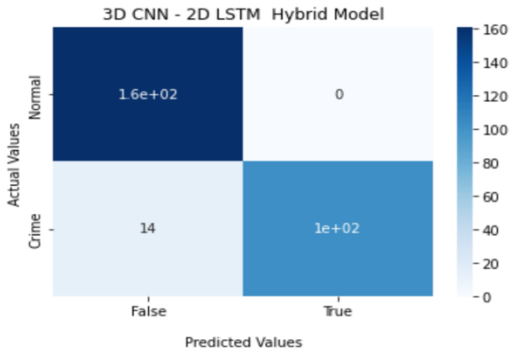


*Figure 21 Report and Confusion Matrix*

### 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 44 minutes for training only without validating. Which is 66 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 algorithm was needed to be used where the frames are added 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 overcome 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 13 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 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.