

TABLE OF CONTENTS

	Page
LIST OF ABBREVIATIONS	2
LIST OF FIGURES	3
LIST OF TABLES	4
ABSTRACT	5
CHAPTER 1 INTRODUCTION.....	6
1.1.	6
1.2.	6
CHAPTER 2 PROBLEM DEFINITION	7
CHAPTER 3 LITERATURE SURVEY.....	8
CHAPTER 4 PROJECT DESCRIPTION.....	8
4.1. PROPOSED DESIGN	9
4.2. ASSUMPTIONS AND DEPENDENCIES.....	10
CHAPTER 5 REQUIREMENTS	11
5.1. FUNCTIONAL REQUIREMENTS	11
5.2.	11
CHAPTER 6 METHODOLOGY.....	12
CHAPTER 7 EXPERIMENTATION.....	14
CHAPTER 8 TESTING AND RESULTS	15
REFERENCES... ..	19
APPENDIX A	

NOMENCLATURE USED

CNN	Convolution Neural Network
UCSD	University of California San Diego
GUI	Graphical User Interface
API	Application Programming Interface
ARFF	Attribute-Relation File Format

LIST OF FIGURES

Fig. No.	Description of the figure	Page No.
1	Machine Learning Flowchart	17
2	Data Set	20
3	Proposed Flow Chart	23
4	Frame containing anomaly	33
5	Video converted into frames	34
6	Resizing each frame & Predicting score	35
7	Final Output	36

LIST OF TABLES

Table No.	Description of the Table	Page No.
1	Crime Rates	10
2	Functional Requirements	26

ABSTRACT

During the last few decades, surveillance cameras have been installed in different locations. Analysis of the information captured using these cameras can play effective roles in event prediction, online monitoring and goal-driven analysis applications including anomalies and intrusion detection. Nowadays, various Artificial Intelligence techniques have been used to detect anomalies, amongst them convolutional neural networks using deep learning techniques improved the detection accuracy significantly. The goal of this article is to propose a new method based on deep learning techniques for Crime detection in video surveillance cameras. The proposed method has been evaluated in the UCSD dataset, and showed an increase in the accuracy of Public Risk Deterrence or Crime detection.

CHAPTER 1

INTRODUCTION

CHAPTER 1 INTRODUCTION

Unusual activity or Crime scene detection with the use of unsupervised machine learning techniques is still an open debate in the field of machine learning. Crime means the occurrence of events or behaviors which are unusual, irregular, unexpected and unpredictable and thus different from existing patterns. Detecting anomalies by learning from normal data can have important and different applications. And also, an Crime detection process is completely dependent on the environment, context and Crime scenario.

In different scenarios, anomalies will accordingly be different. Existing supervised methods for Crime detection such as simple CNN based methods require labels which are difficult to attain due to the video high dimension information. High dimension of video affects representation and creation of a model. In this project, Crime detection is based on videos of surveillance cameras. It should be noted that detection in videos is more difficult than in other data since it involves detection methods and also requires video processing as well. With the increase in threat to public safety, in the last decade, different observation cameras have been introduced in various areas to guarantee the well-being of individuals.

Security is a tedious, complicated and tough job in today's digitized world. Various anomaly detection techniques have been brought forward to design a smart control system. Observation information is produced through the camera, and then this information is transmitted over the system to the storage. Examination of the data caught utilizing these cameras can assume dynamic jobs in occasion forecast, Internet checking and objective-driven investigation applications, including oddities and interruption location.

Various AI methods are being used today that can detect abnormalities with ease, in particular, the convolutional neural networks that make use of deep learning techniques have proven to increase the accuracy of crime detection significantly. The objective of this paper is to propose another strategy dependent on profound learning methods for wrongdoing recognition in video reconnaissance cameras. The proposed strategy has been assessed in the UCSD dataset and indicated an expansion in the exactness of public risk deterrence or crime identification.

1.1. TECHNIQUES:

This project introduces an Crime detection method based on deep learning techniques. The architecture of this method has two main phases which are called train network and detection classifier. The first phase aims for feature extraction and is consisted of five components with a deep structure. The aim of the second phase is detection. This phase is consisted of five deep neural network classifiers and reconstruction network. Each component in detection phase produces a detected class and a score. At last, by these detection classes and scores, the ensemble classifier performs the final detection and announces it.

1.1.1. ADVANTAGES:

The processing of surveillance cameras information in crowded scenes poses serious challenges and difficulties. If this process is online, the complexity will even increase. One of the best approaches for processing this information and consequently achieving the goal-oriented pattern is the use of advanced machine learning techniques such as deep learning approaches. The advantage of these types of processes, which usually have a high dimensional data, can be traced back to the existence of an end-to-end system. End-to-end systems automate feature extraction. One of the main purpose of using deep learning is to extract information from high dimension data.

CHAPTER 2

PROBLEM DEFINITION

CHAPTER 2 PROBLEM DEFINITION

Existing supervised methods for Crime detection such as simple CNN based methods require labels which are difficult to attain due to the video high dimension information. High dimension of video affects representation and creation of a model. In this project, Crime detection is based on videos of surveillance cameras. It should be noted that detection in videos is more difficult than in other data since it involves detection methods and also requires video processing as well.

Problem: “Lets Say a man is caught being stabbed by a gangster on a road and is about to die.”

Solution: “The video surveillance camera detects the anomaly and sends a notification to a nearby emmergency and the person can be saved.”

In this project, Crime detection is based on videos of surveillance cameras where the video will be converted into frames and detection is achieved.

2.1. Crime Rates:

The table below shows the statistics of the crime rates in India and major states having crime rates.

Country/State	2016	2017	2018	2019	Percentage Share
India	4831515	5007044	5074635	5156172	100.0%
Karnataka	179479	184063	163416	163691	3.2%
Kerala	707870	653500	512167	453083	8.8%

Table 1 :- Ref: [Crime in India - Wikipedia](#)

By the above table (Table 1) we can see that the total crime rate by the year 2019 in India was 5156172 crimes. Kerala and Karnataka alone holds the share of 8.8% and 3.2% respectively. We know its difficult to decrease the crime rate but atleast we can try to decrease the death rate by our research.

2.2. OBJECTIVES:

1. Divide video into frames and divide test frames to defined patches.
2. Train the model to analyze the test data and detect the Public Risk Deterrence or threatening scene.
3. Send SMS to the control station about detection of abnormal event.

CHAPTER 3

LITERATURE REVIEW

CHAPTER 3 LITERATURE REVIEW

Due to the existence of rich and analytical information in videos and their easy accessibility, scientific researchers have been interested in the analysis and processing of these kinds of data. One of the challenges in analyzing video data is objects detection in video frames. Also, video crime detection has been one of the controversial research topics within the recent years. In the last few years, deep learning approaches have also been introduced for the implementation of crime detection methods. In all crime detection approaches, learning is achieved solely through normal data. Another important point regarding the anomalies is that abnormal events are usually rare events that occur comparatively less than other normal incidents.

The purpose of this literature review was to compile recent conducted reviews about the use of artificial intelligence (AI) in crime prediction and prevention. How capable are current AI-technologies in predicting crime? How capable are current AI- technologies in preventing crime? The results suggested that the currently AI technologies in use were capable of predicting and preventing crime. They could find patterns in large data sets in much more efficient manner than humans could. The most commonly used AI technologies in crime prediction were data mining, and machine learning and deep learning were most commonly used in crime prevention. The results of this literature review indicated that the use of AI systems in crime prediction and prevention is growing, however it can still be viewed as a new phenomenon that requires further research. To achieve higher levels of accuracy there was requirements for access to larger data sets and further testing and training of the AI models. Criminologists also needs to pay more attention to the field of AI in crime prediction and prevention so that they can direct the practice.

When discussing about applying AI in criminal cases it can be mainly classified into two areas as AI for crime detection and AI for crime prevention.

A. *AI for crime detection.*

i) Public safety videos and images:

Video and image analysis is used in criminal justice and law enforcement communities to obtain information about people, objects, and actions to support criminal investigations. However, the analysis of video and image data requires a lot of staff, requiring significant investment in the knowledgeable staff of the story. Video and image analysis is also prone to human error due to the abundance of information, the rapid change of technology such as smartphones and apps, and the limited number of specialized staff with experience in processing that information(“Assistive AI keeps the human element in public safety | 2021-06-04 | Security Magazine,” n.d.). AI technology provides the ability to overcome such human mistakes and to act as an expert. Traditional software algorithms that help people are limited to predetermined factors such as eye shape, eye color, and the distance between eyes to see face or human details for pattern analysis. Video algorithms and image AI not only learn complex tasks but also develop and determine

the complexities / limitations of their complex facial expressions to perform these tasks, more than people can imagine.

B. AI for crime prevention.

i)Predicting the crime spots:

Imagine a thief coming to his next heist to find out that the police are already waiting for him. Yes, it can be done using AI technology. AI programming, along with big data, can help identify crime hotspots. Crime types often interact with space and time, and crime-related information such as crime type, crime scene, and crime weapons can help predict future crime scenes. For example, an outbreak of theft in one area may help predict that similar incidents may occur in the surrounding area in the future. AI programs can help police find a place where they should consider extra vigilance.

3.1 CHALLENGES:

The challenges for detecting anomalies in videos include speed, online alerts, and localization. It should be mentioned that crime localization is very crucial and most of the existing systems and data lack it. In some approaches, the localization is performed in the pre-processing step which is usually based on video frames comparison. This will increase the accuracy . In other words, most of the existing approaches and available datasets only indicate the presence of anomalies and do not specifying their location. The current methods also lack appropriate training data and correct crime description along with their high cost of extracting features which directly affect detection.

CHAPTER 4

PROJECT DESCRIPTION

CHAPTER 4 PROJECT DESCRIPTION

The proposed method of this project is based on deep learning techniques for detecting anomalies in video. Two main components are considered for this method. The first component is the extraction and learning of the feature and the second component is the detection of anomalies. Apart from these two components, there is a pre-processing step which is related to background estimation and removal. Like all machine learning approaches, this method also has two main train phase and test phase. In train phase, features are trained by train parts of dataset which contains only normal frames, and trained model in test phase is used by other parts of dataset which contain abnormal frames.

One of the widely used methods for detecting anomalies is the use of a binary classifier which has two normal and abnormal classes. The normal class contains data whose occurrence frequency is high, while the other class contains rare and unseen events in accordance with the data pattern.

Like other machine learning methods, deep learningbased crime detection techniques can also be divided into three categories of supervised, unsupervised and semi-supervised. Supervised crime detection needs labeled data which is difficult because of the volume and dimension of data. In a supervised approach, the main operation is decision rules based or model based which can distinguish between two classes. And also unsupervised methods need complex computing. Unsupervised methods are also known as data driven crime detection.

Artificial intelligence is the capability of a computer system to mimic human cognitive functions such as learning and problem-solving. Through AI, a computer system uses maths and logic to simulate the reasoning that people use to learn from new information and make decisions.

Machine learning is an application of AI. It's the process of using mathematical models of data to help a computer learn without direct instruction. This enables a computer system to continue learning and improving on its own, based on experience. One way to train a computer to mimic human reasoning is to use a neural network, which is a series of algorithms that are modeled after the human brain. The neural network helps the computer system achieve AI through deep learning. The complete structure of machine learning project using AI is shown in below flowchart (Fig1).

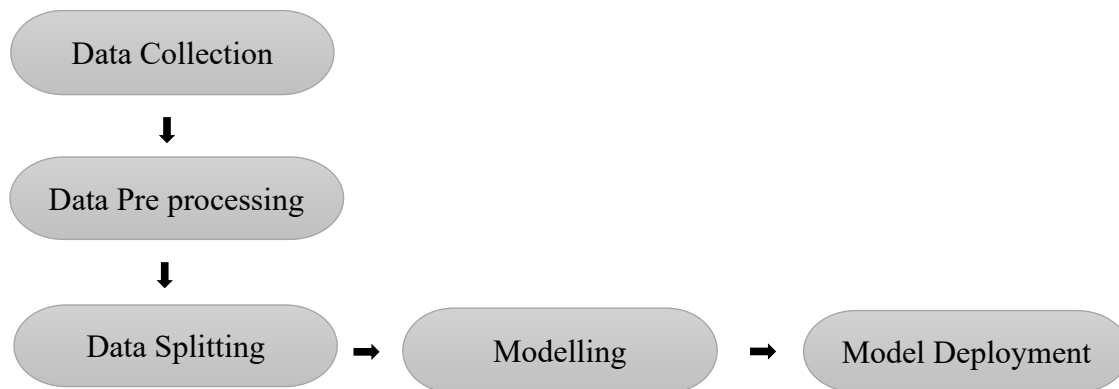


Fig 1:- ML Flowchart

A. *Data Collection*

It's time for a data analyst to pick up the baton and lead the way to machine learning implementation. The job of a data analyst is to find ways and sources of collecting relevant and comprehensive data, interpreting it, and analyzing results with the help of statistical techniques. The type of data depends on what you want to predict. There is no exact answer to the question "How much data is needed?" because each machine learning problem is unique. In turn, the number of attributes data scientists will use when building a predictive model depends on the attributes' predictive value.

'The more, the better' approach is reasonable for this phase. Some data scientists suggest considering that less than one-third of collected data may be useful. It's difficult to estimate which part of the data will provide the most accurate results until the model training begins. That's why it's important to collect and store all data — internal and open, structured and unstructured.

B. *Data Preprocessing*

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data preprocessing task.

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

It involves below steps:

1. Getting the dataset

2. Importing libraries
3. Importing datasets
4. Finding Missing Data
5. Encoding Categorical Data
6. Splitting dataset into training and test set
7. Feature scaling

Getting the dataset

To create a machine learning model, the first thing we required is a dataset as a machine learning model completely works on data. The collected data for a particular problem in a proper format is known as the dataset. Dataset may be of different formats for different purposes, such as, if we want to create a machine learning model for business purpose, then dataset will be different with the dataset required for a liver patient. So each dataset is different from another dataset. To use the dataset in our code, we usually put it into a CSV file. However, sometimes, we may also need to use an HTML or xlsx file. CSV stands for "Comma-Separated Values" files; it is a file format which allows us to save the tabular data, such as spreadsheets.

Importing Libraries

In order to perform data preprocessing using Python, we need to import some predefined Python libraries. These libraries are used to perform some specific jobs. There are three specific libraries that we will use for data preprocessing, which are:

Numpy: Numpy Python library is used for including any type of mathematical operation in the code. It is the fundamental package for scientific calculation in Python. It also supports to add large, multidimensional arrays and matrices.

Matplotlib: The second library is matplotlib, which is a Python 2D plotting library, and with this library, we need to import a sub-library pyplot. This library is used to plot any type of charts in Python for the code.

Pandas: The last library is the Pandas library, which is one of the most famous Python libraries and used for importing and managing the datasets. It is an open-source data manipulation and analysis library.

Importing datasets

Now we need to import the datasets which we have collected for our machine learning project. But before importing a dataset, we need to set the current directory as a working directory. To set a working directory in Spyder IDE, we need to follow the below steps:

1. Save your Python file in the directory which contains dataset.
2. Go to File explorer option in Spyder IDE, and select the required directory.

3. Click on F5 button or run option to execute the file.

Now to import the dataset, we will use `read_csv()` function of pandas library, which is used to read a csv file and performs various operations on it. Using this function, we can read a csv file locally as well as through an URL.

Finding Missing Data

The next step of data preprocessing is to handle missing data in the datasets. If our dataset contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset.

Ways to handle missing data:

There are mainly two ways to handle missing data, which are:

By deleting the particular row: The first way is used to commonly deal with null values. In this way, we just delete the specific row or column which consists of null values. But this way is not so efficient and removing data may lead to loss of information which will not give the accurate output.

By calculating the mean: In this way, we will calculate the mean of that column or row which contains any missing value and will put it on the place of missing value. This strategy is useful for the features which have numeric data such as age, salary, year, etc. Here, we will use this approach.

Encoding Categorical Data

Since machine learning model completely works on mathematics and numbers, but if our dataset would have a categorical variable, then it may create trouble while building the model. So it is necessary to encode these categorical variables into numbers.

Splitting the Dataset into the Training set and Test set

In machine learning data preprocessing, we divide our dataset into a training set and test set. This is one of the crucial steps of data preprocessing as by doing this, we can enhance the performance of our machine learning model.

Suppose, if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models.

If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset.

Here, we can define these datasets as:

Training Set (80%)	Testing Set (20%)
--------------------	-------------------

Fig 2 :- Dataset

Training Set: A subset of dataset to train the machine learning model, and we already know the output.

Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

Feature Scaling

Feature scaling is the final step of data preprocessing in machine learning. It is a technique to standardize the independent variables of the dataset in a specific range. In feature scaling, we put our variables in the same range and in the same scale so that no any variable dominate the other variable.

C. Data Splitting

In machine learning, data splitting is typically done to avoid overfitting. That is an instance where a machine learning model fits its training data too well and fails to reliably fit additional data.

The original data in a machine learning model is typically taken and split into three or four sets. The three sets commonly used are the training set, the dev set and the testing set:

1. The **training set** is the portion of data used to train the model. The model should observe and learn from the training set, optimizing any of its parameters.

2. The **dev set** is a data set of examples used to change learning process parameters. It is also called the *cross-validation* or *model validation set*. This set of data has the goal of ranking the model's accuracy and can help with model selection.
3. The **testing set** is the portion of data that is tested in the final model and is compared against the previous sets of data. The testing set acts as an evaluation of the final mode and algorithm.

Data should be split so that data sets can have a high amount of training data. For example, data might be split at an 80-20 or a 70-30 ratio of training vs. testing data. The exact ratio depends on the data, but a 70-20-10 ratio for training, dev and test splits is optimal for small data sets.

D. Modeling

Model optimisation is an integral part of achieving accuracy in a live environment when building a machine learning model. The aim is to tweak model configuration to improve accuracy and efficiency. Models can also be optimised to fit specific goals, tasks, or use cases. Machine learning models will have a degree of error, and optimisation is the process of lowering this degree.

The process of machine learning optimisation involves the assessment and reconfiguration of model hyperparameters, which are model configurations set by the data scientist.

Hyperparameters aren't learned or developed by the model through machine learning. Instead, these are configurations chosen and set by the designer of the model. Examples of hyperparameters include the structure of the model, the learning rate, or the number of clusters a model should categorise data into. The model will perform its tasks more effectively after optimisation of the hyperparameters.

Historically, the process of hyperparameter optimisation may have been performed through trial and error. This would be extremely time consuming and resource intensive. Now, optimisation algorithms are used to rapidly assess hyperparameter configuration to identify the most effective settings. Examples include bayesian optimisation, which takes a sequential approach to

hyperparameter analysis. It takes into account hyperparameter's effect on the target functions, so focuses on optimising the configuration to bring the most benefit.

E. Model Deployment

The last step in building a machine learning model is the deployment of the model. Machine learning models are generally developed and tested in a local or offline environment using training and testing datasets. Deployment is when the model is moved into a live environment, dealing with new and unseen data. This is the point that the model starts to bring a return on investment to the organisation, as it is performing the task it was trained to do with live data.

More and more organisations are leveraging containerisation as a tool for machine learning deployment. Containers are a popular environment for deploying machine learning models as the approach makes updating or deploying different parts of the model more straightforward. As well as providing a consistent environment for a model to function, containers are also intrinsically scalable. Open-source platforms like Kubernetes are used to manage and orchestrate containers, and automate elements of container management like scheduling and scaling.

Products such as Seldon Deploy are used to streamline model deployment and management. It's a language-agnostic platform which integrates a deployed model with other apps through API connections. It also provides workflow management tools for a streamlined deployment process, and an analytics dashboard to monitor the health of the model. Once deployed, it's vital that the model is continuously monitored for model drift so that it stays accurate and effective.

4.1 PROPOSED DESIGN:

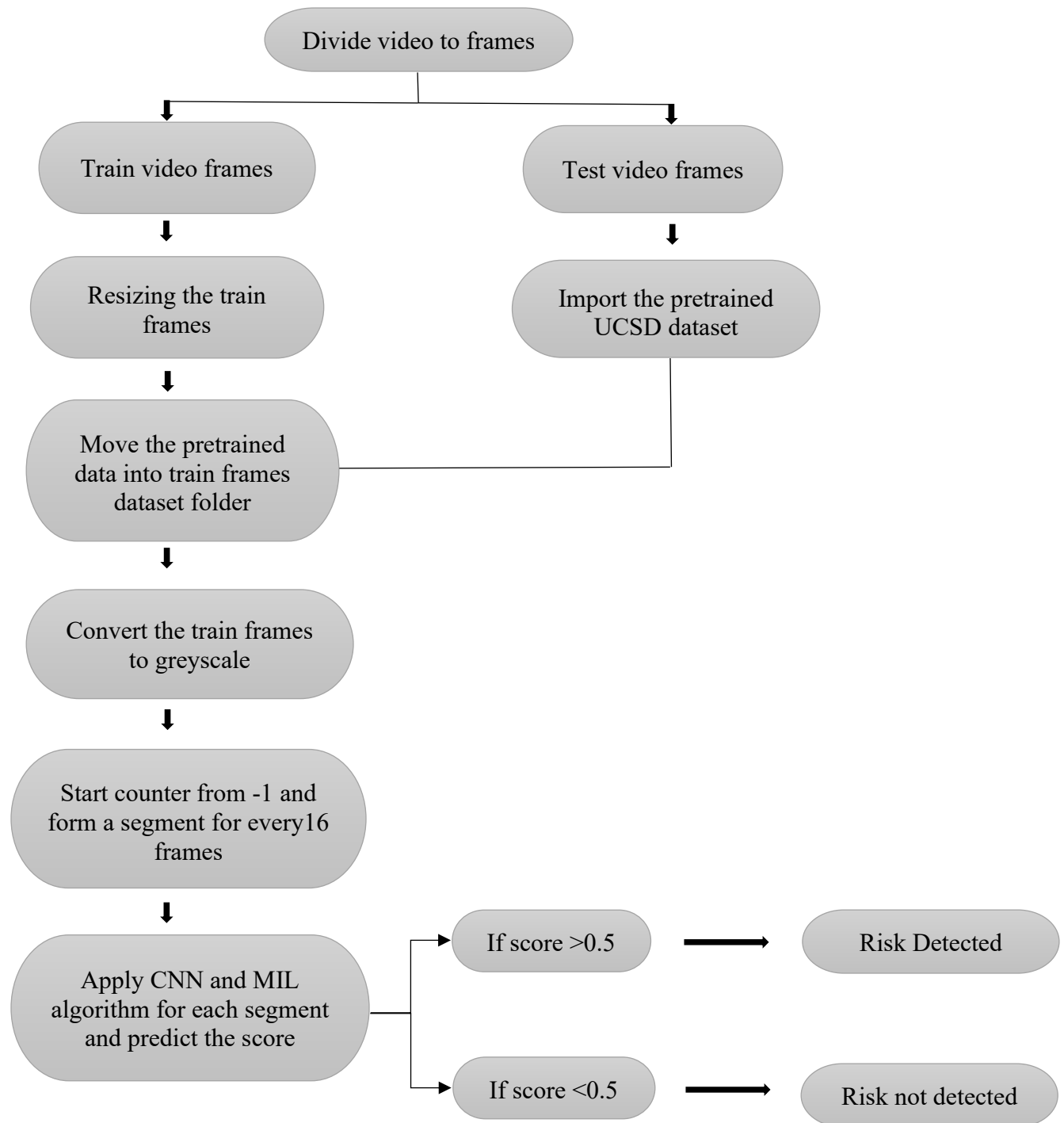


Fig 3 :- Flowchart

4.2 ASSUMPTIONS AND DEPENDIES

As can be seen in the design, learning features are of four main types. For some types, feature extraction processes are performed on single frames, and others are based on patch frames in order to reduce cost and training time. The first feature is appearance which is related to object detection in each frame; and by comparing each frame with previous and next frames the detection score is generated.

The second feature is density which is about density of objects in each frame; the final score is generated based on frames comparison and average speed. The third feature is motion which is based on the flow of objects between patch frames and it generates optical flow and a sequence of video then used for another score on anomaly.

The last feature is scene which is based on patch frames and reconstructing a scene from learned model. The combination of these features is also used for detection and creation of scores.

CHAPTER 5

REQUIREMENTS

CHAPTER 5 REQUIREMENTS

5.1 FUNCTIONAL REQUIREMENTS:

Functional Requirement No	Functional Requirement Description
FR1	The user should able to upload the video containing anomaly or non anomaly
FR2	The UI Component should accept the data uploaded.
FR3	The API should redirect to backend by connecting through port number.
FR4	An update notification has to be sent through SMS portal.
FR5	The UI displays the final outcome of the video uploaded.

Table 2 :- Functional Requirement

5.2 SOFTWARE REQUIREMENTS:

- Visual Studio
- Anaconda(Python3)
- Tensorflow

5.3 HARDWARE REQUIREMENTS:

- 1 GHz processor or faster 32-bit (x86) or 64-bit (x64).
1 GB of RAM for 32-bit or 2 GB of RAM for 64-bit.
- 16 GB of hard drive space for 32-bit or 20 GB for 64-bit

CHAPTER 6

METHODOLOGY

CHAPTER 6 METHODOLOGY

A. *Pre-Processing*

The first step before starting extracting and learning features is to estimate and remove the background. The background is indeed different for different scenarios as there are various methods for its removal. For instance, the background might include empty spaces or street borders.

In this method, the background estimation is based on most occurrence of frequency (MOF) between video frame patches [9]. For the background estimation steps at first, a histogram is generated for each frame of the video which is based on pixels and their location in the image

Then the histogram of the frames in each patch is compared with each other, and the maximum values per patch are identified as background and are thus grayed. Removing the background will reduce the cost of the computing and the processing time. This step is considered as a part of train network.

B. *Feature Extraction and Learning Component*

In addition to background estimation, train network has four main components. The deep network for extracting appearance feature uses a stacked denoising auto-encoder (SDAE) with 6 encode layer and the same structure of decode layer. Each frame is convolving to network with 1×1 window size and it includes stride and padding. All frames normalize in binary mode. This SDAE has 6 encode layers and 6 same structure in decode layer which is deeper than the existing methods.

The output of this step is detected objects which are called appearance representation.

This output is used in detecting phase and also is utilized as an input to density estimation component in order to increase the accuracy of estimation. Density Estimation is carried out by convolutional neural network with 8×8 . Windows filter. The output of this component is feature map and the loss function is computed based on square error. In the estimation of the density, the sectors associated with the background are considered zero.

The third component is motion feature extractor. It performs a feature extraction based on the direction of moving objects in the scene of video patches. This deep network also has a similar structure to appearance feature extractor but it is based on frames patches. After entering the patch frame into the network, computing optical flow will be done based on comparison of frames in a patch. The output of this step is Motion Representation which is used for future detection.

The last component is Scene Reconstruction which is based on reconstruction network. The structure of this reconstruction network is based on convolutional Auto- Encoder with the same

CNN generator and discriminator networks. Generator part regenerate the scene which has 10 layers to reconstruct frames based on the previous and the next frame in same patch and the discriminator compares the generated scene with original one in order to compute the reconstruction error. It should be mentioned that discriminator part has the same structure as that of the generator. A high reconstruction error during test indicates anomalies. The reconstruction error in train network is low and this will be a measure for detecting anomalies.

At the end of the training step, a set of learned and combined features is created in order to achieve Crime detection.

C. Detection Component

In the detection component, learned features which are generated in train network are given to a classifier with two classes of normal and abnormal. Features are given as individual and combined feature to these networks.

Reconstruction error and appearance features are given to network as a combined feature since the appearance feature or object detection with a reconstruction error can be a strong feature for the detection of anomalies. The lower reconstruction error for the corresponding frame will make the detection more accurate.

Two other combination features are Motion Feature and density map. These are two complementary features and the direction of motion must be equal to the transfer of density direction.

CHAPTER 7

EXPERIMENTATION

CHAPTER 7 EXPERIMENTATION

In this paper, a new deep learning based for Crime detection of video surveillance cameras is introduced. One advantage of this method is the use of deep learning techniques in all train and detection components. The two main components of this method are evaluated based on some metrics and with UCSD dataset which is the most famous crime detection dataset. Another benefit of this method is the isolation of train network phase.

So it can use as a pre-train Network in similar works. For further improvement, it is possible to add a component which can add descriptions to each detection classifier or to the last one; or it is possible to add a component in the detection phase which can localize the crime accurately.

CHAPTER 8

TESTING AND RESULTS

CHAPTER 8 TESTING AND RESULTS

8.1 One of the frame of a video containing anomaly:



Fig 4:- Frame of a video

In the above figure we can see one of the frame of a video containing risk in it. The video once after it gets uploaded it gets converted into frames. We can upload any video but it has to be in an mp4 format. Our project is designed in a such a way that it takes more of a cctv footages rather than hd videos. We can easily upload these video in our web page created using Flask. The user interface is created in such way that we can see upload button, predict button and even the final output on a single interface.

8.2 Video converted into Frames after uploading:

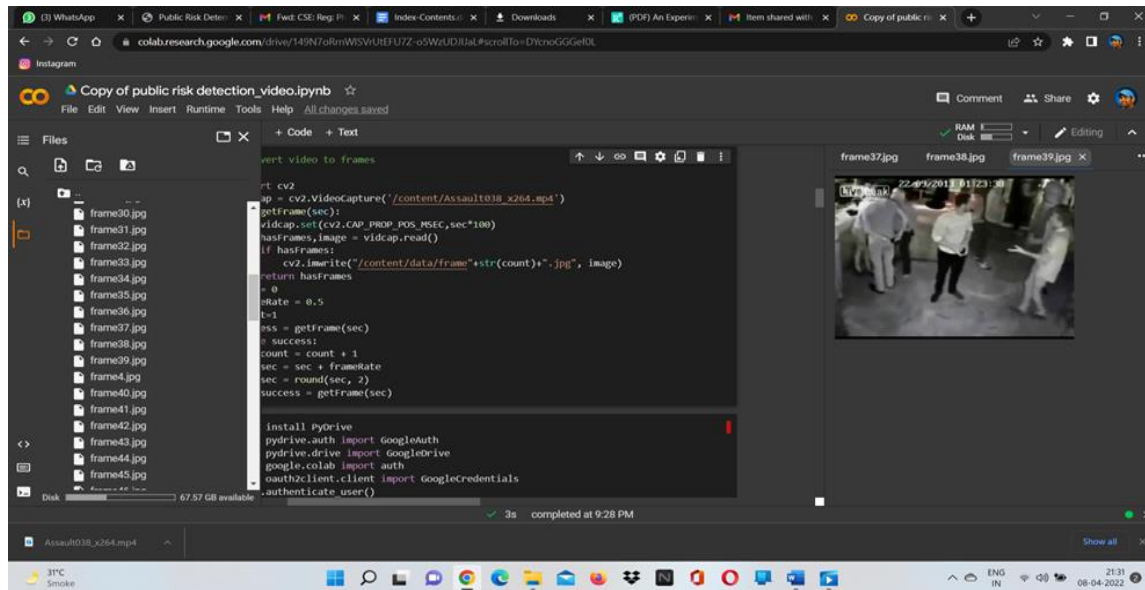
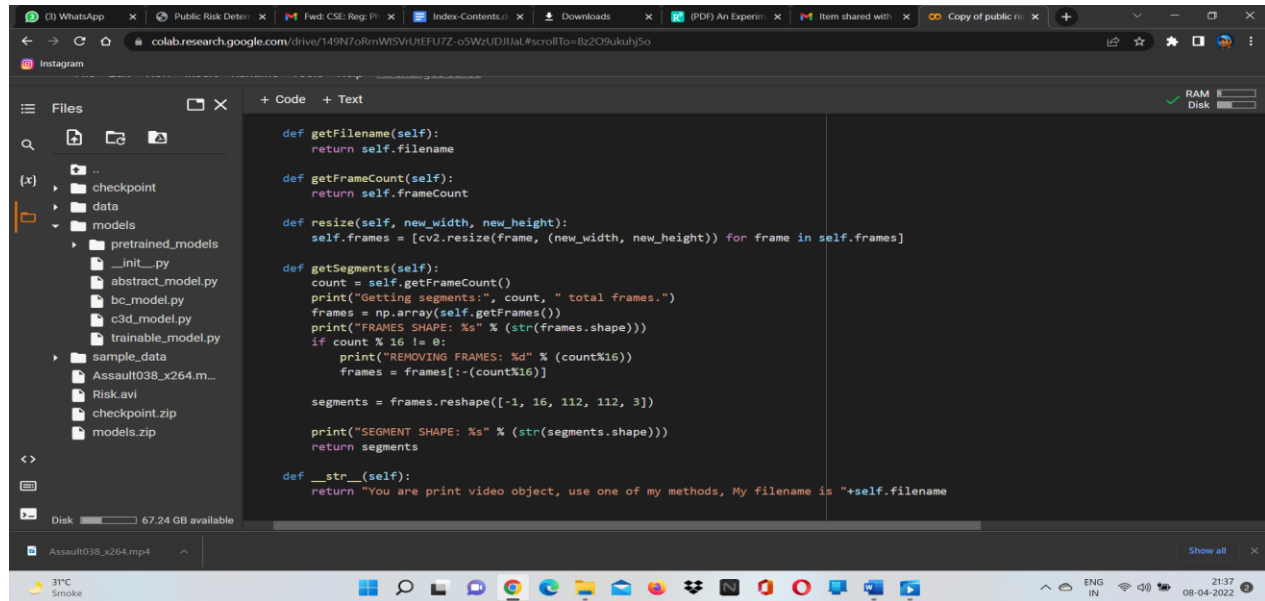


Fig 5:- One of the anomaly frame

In the above figure we can see one of the frame of a video containing risk in it. The video once after it gets uploaded it gets converted into frames. Now these frames divide themselves into training and testing. Under training all the frames of a video gets stored in folder and gets trained. The above figure (fig 5) is one of the frame from training dataset. In order to convert these video into frames we reduce the frames per second count and store each frames in a separate folder

8.3 Function to resize each frame and predict the score of each frame:



```
def getFilename(self):
    return self.filename

def getFrameCount(self):
    return self.frameCount

def resize(self, new_width, new_height):
    self.frames = [cv2.resize(frame, (new_width, new_height)) for frame in self.frames]

def getSegments(self):
    count = self.getFrameCount()
    print("Getting segments:", count, " total frames.")
    frames = np.array(self.getFrames())
    print("FRAMES SHAPE: %s" % (str(frames.shape)))
    if count % 16 != 0:
        print("REMOVING FRAMES: %d" % (count%16))
        frames = frames[:-(count%16)]

    segments = frames.reshape([-1, 16, 112, 112, 3])

    print("SEGMENT SHAPE: %s" % (str(segments.shape)))
    return segments

def __str__(self):
    return "You are print video object, use one of my methods, My filename is "+self.filename
```

Fig 6 :- Function to resize all the frames

After the clip gets converted into frames now these frames will be in irregular shapes of their own. In order to run the process in an efficient way we resize all the frames to a standard size. Once the frames get converted to standard size the pretrained dataset gets moved to the training frames dataset folder. These pretrained dataset helps to extract the features from the training dataset. There will be a background estimation in the frame once it gets converted to the grey scale. Now all features of the foreground will be extracted.

8.4 Final output displaying the result as Risk Detected:

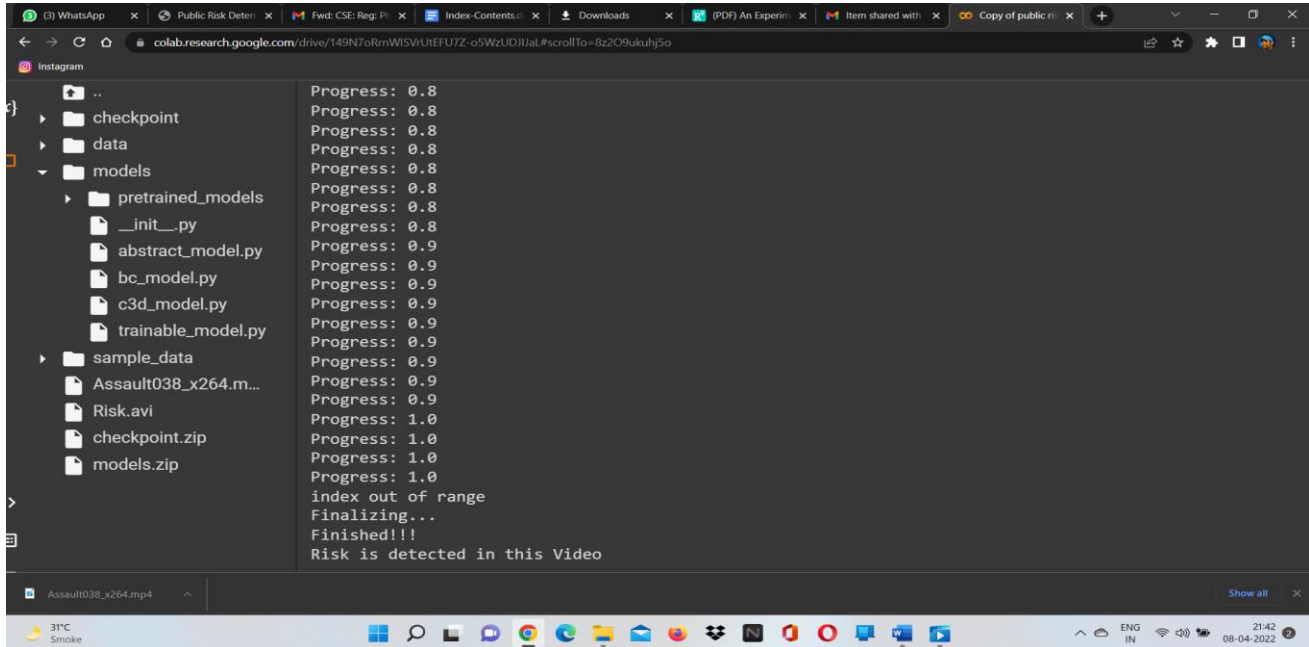


Fig 7:- Final output displaying risk detected

After the focus on the foreground of the frames. The counter starts its pointer from -1 to n frames of the video. Now for every 16 frames it gets converted into a segment upto nth frame. This makes the process efficient. For all the segments of the video we apply CNN and Multiple Instance Learning Algorithm. These algorithms predicts the score of each segment. If the score exceeds 0.5 then it considers that segment as anomaly and hence the uploaded video has risk in it. If the score is less than 0.5 then it considers as Non Anomaly video. Now this video gets attached to the senders mail ID and sends the video to the senders mail ID. If the video has risk in it then there will be a label displayed in the attached video as “Risk Detected”.

REFERENCES

REFERENCES

- [1] Dinesh Kumar Saini, Dikshika Ahir and Amit Ganatra, “Techniques and Challenges in Building Intelligent Systems: Anomaly Detection in Camera Surveillance”, Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems, Springer International Publishing Switzerland, 2016
- [2] B Ravi Kiran, Dilip Mathew Thomas, Ranjith Parakkal, “An overview of deep learning based methods for unsupervised and semisupervised anomaly detection in videos”, MDPI Journal of Imaging, arXiv:1801.03149v1, 2018
- [3] Ryota Hinami, Tao Mei, and Shin’ichi Satoh, “Joint Detection and Recounting of Abnormal Events by Learning Deep Generic Knowledge”, arXiv:1709.09121v1, 2017
- [4] M. Ribeiro, A.E.L., and H. S. Lopes, “A study of deep convolutional auto-encoders for anomaly detection in videos”, Pattern Recognition Letters, ELSEVIER, 2017
- [5] Yong Shean Chong, Yong Haur Tay, “Abnormal Event Detection in Videos Using Spatiotemporal Autoencoder”, International Symposium on Neural Networks, Springer International Publishing AG, 2017
- [6] Hung Vu, “Deep Abnormality Detection in Video Data”, Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, 2017
- [7] M. Sabokroua, M.F., M. Fathyc, Z. Moayeddd and R. Kletted, “Deep- Anomaly: Fully Convolutional Neural Network for Fast Anomaly Detection in Crowded Scenes”, Journal of Computer Vision and Image Understanding, 2017

- [8] Qaisar Abbas, Mostafa E. A. Ibrahim, M. Arfan Jaffar¹, “Video scene analysis: an overview and challenges on deep learning algorithms”, Multimedia Tools and Applications, Springer, 2017
- [9] Revathi, A. R., Kumar, Dhananjay “An efficient system for anomaly detection using deep learning classifier”, Signal, Image and Video Processing, Springer, 2016
- [10] Hung Vu, Tu Dinh Nguyen, Anthony Travers, Svetha Venkatesh And Dinh Phung, “Anthony Travers, Energy-Based Localized Anomaly Detection in Video Surveillance”, Springer International Publishing AG, 2017
- [11] Siqi Wanga, E.Z., Jianping Yin, “Video anomaly detection and localization by local motion based joint video representation and OCELM”, Neurocomputing, 2017
- [12] M. Sabokrou, M. Fathy, M. Hoseini., “Video anomaly detection and localisation based on the sparsity and reconstruction error of autoencoder”, ELECTRONICS LETTERS, IEEE, 2016.
- [13] Yong Shean Chong, Yong Haur Tay, “Modeling Video-based Anomaly Detection using Deep Architectures: Challenges and Possibilities”, Control Conference (ASCC), IEEE, 2015
- [14] Shean Chong, Yong Haur Tay, Yong, “Modeling Representation of Videos for Anomaly Detection using Deep Learning: A Review”, arXiv:1505.00523v1, 2015.