SYSTEM FOR IDENTIFYING AGGRESSIVE BEHAVIOURS IN PUBLIC PLACES USING CNN MODEL

A PROJECT REPORT

Submitted By

BHUVANESH KUMAR S	**[211419104042]**
DINESH KUMAR M	**[211419104069]**
DEVESH R	**[211419104056]**

In partial fulfilment for the award of the degree

Of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2023

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this mini project report “**SYSTEM FOR IDENTIFYING AGGRESSIVE BEHAVIOURS IN PUBLIC PLACES USING CNN MODEL”** is the bonafide work of **“S BHUVANESH KUMAR (211419104042) M DINESH KUMAR (211419104069) R DEVESH (211419104056)”** who carried out the project work under my supervision.

SIGNATURE **Dr. L.JABASHEELA , M.E.,Ph.D.,** **HEAD OF THE DEPARTMENT** DEPARTMENT OF CSE, PANIMALAR ENGINEERING COLLEGE, NASARATHPETTAI, POONAMALLEE, CHENNAI-600 123.	**SIGNATURE** **Dr.N.PUGHAZENDI,M.E.,Ph.D.,** **PROFESSOR** DEPARTMENT OF CSE, PANIMALAR ENGINEERINGCOLLEGE, NASARATHPETTAI, POONAMALLEE, CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the **End Semester** Project Viva-Voce Examination held on...........................

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENTS

We **S BHUVANESH KUMAR (211419104042), M DINESH KUMAR (211419104069), R DEVESH (211419104056)** hereby declare that this project report titled **"SYSTEM FOR IDENTIFYING AGGRESSIVE BEHAVIOURS IN PUBLIC PLACES USING CNN MODEL"**, under the guidance of **Dr.N.PUGHAZENDI,M.E., Ph.D.,** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by me.

BHUVANESH KUMAR S
DINESH KUMAR M
DEVESH R

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr. P. CHINNADURAI, M.A., Ph.D**. for his kind words and enthusiastic motivation, which inspired me a lot in completing this project.

We express our sincere dedication and thanks to our beloved Directors **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D**. and **Dr. SARANYASREE SAKTHI KUMAR B.E., M.B.A., Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr. K. MANI, M.E., Ph.D**. who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. L.JABASHEELA , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank our parents, friends, project Guide and coordinator **Dr.N.PUGHAZENDI,M.E.,Ph.D.,** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

BHUVANESH KUMAR S
DINESH KUMAR M
DEVESH R

ABSTRACT

Detection of unusual human behavior in public places has been a tedious process if we doit as manual process. As it is an important task, human process may be random and unpredictable and classification of suspicious behavior of human can be very difficult. So,we are proposing a system which works as an Automated Surveillance System to detect and track suspicious human behavior. Automated surveillance allows for the monitoring of human activity in sensitive and well-known places including bus stops, train stations, airports, banks, shopping centers, schools, colleges, parking lots, and roadways in order to prevent terrorism, theft, accidents, illegal parking, fighting, chain snatching, crime, and other shady activities. All the procedures used to identify human activity in surveillance videos have generally been covered in the literature. These procedures include foregroundobject extraction, object detection using feature extraction, activity analysis, and recognition. In our project, we have decided to overcome the drawbacks of the existing systems by using CNN (Convolutional Neural Networks) Model. The primary goal of video surveillance is to replace the current passive version so that aberrant human activitymay be recorded and, after analysis, an alert can be generated through alarms to stop odd activity.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS, ABBREVIATIONS

AI	Artificial Intelligence
YOLO	You Only Look Once
CNN	Convolutional Neural Network
OpenCV	Open Source Computer Vision
RCNN	Regions Convolutional Neural Networks
UML	Unified Modelling Language
GPU	Graphical Processing Unit
CCTV	Closed-Circuit Television
ATM	Automatic Teller Machine
KLT	Kanade-Lucas-Tomasi
SARD	Suspicious Activity Region Detector
ERD	Entity Relationship diagram
DFD	Data flow diagram

CHAPTER 1
INTRODUCTION

1. INTRODUCTION

1.1 OVERVIEW

With rising crime rates, it becomes a concern if criminals are not recognized in a timely manner and the appropriate precautionary measures are not taken. There are surveillance systems deployed in the majority of urban and metropolitan regions, which are constantly collecting data. There is a greater likelihood of suspicious activity occurring as a result of the massive accumulation of surveillance data. The rate of crime or violence has been rising globally in recent years. Several tools are employed to lessen or regulate the issue. The finest alternative that may be used in both public and private locations is video surveillance. When the video surveillance system efficiently spots any suspicious or anomalous activity, it is considered to be effective. The majority of today's monitoring systems are run by people. So, they need constant human supervision to spot any unusual behaviour. As a result of human involvementand weariness over time, the system's efficiency drops.

Video surveillance automation can helpto tackle this issue. The automated system's job is to provide notification when a predetermined abnormal activity occurs via alarm. In “Suspicious Activity Detection in Surveillance Footage” by Sathyajit Loganathan and Gayashan Kariyawasam, focuses on the analysis of two cases, including abandoned luggage and crimes involving guns that were captured on surveillance footage. In this method, computer vision and machine learning pipelines are used to detect abandoned luggage in videos and handguns in photos. Instead the concept of this paper are as follows we need to detect person behavior as suspicious or not using CCTV footages and then we have to implement feature extraction concept. The motion features between the two/different objects are extracted to detect the behaviour. Then we can test a video by uploading the frames of the video and then the model predicts its behaviour. This concept has been used in order to recognize and identify the anonymous activity detection in a socially crowded place.

1.2 PROBLEM DEFINITION

Most countries are implementing accurate anomaly detection systems as a means of advancing towards a secure environment due to rising crime rates and general human insecurity worldwide. They are extensively utilised in event detection, health monitoring, fraud detection, fault detection, and ecosystem disturbance detection systems. Considering that the Indian Crime Index is 42.38, the use of anomaly detection technologies is urgently needed. CCTV installations cannot stop crimes like abuse, burglary, explosions, accidents, shooting, and theft on their own. By implementing anomaly detection systems, they must be transformed into intelligent and effective systems. These methods can aid in both identification and the prediction of unexpected activity thanks to their improved versions.

Major set of objectives are as below:

- To identify suspicious human activities in surveillance videos: Robbery, Fight and fire etc. from images, videos and CCTV by using deep learning algorithms.
- To develop Graphical User Interface or a smart phone friendly application.
- To enhances the security of the society by predicting the unusual scenarios and reduce human efforts.

We employed a method that required prior understanding of the Python programming language and machine learning, where we used CNN, to develop this automated system. In light of the project's criteria, it was necessary to list all suspicious human behaviours and work towards their detection of primary activities.

CHAPTER 2

LITERATURE SURVEY

2. LITERATURE SURVEY

2.1 Suspicious Activity Recognition in Video Surveillance System

Author Name : Ms. U. M. Kamthe and Dr. C. G. Patil

Year of Publish : 2018

The project predicts the suspicious activities using some predefined conditions and eliminates the training of Dataset in machine learning methods main focus on ATM loitering and abandoned luggage. The hierarchical technique is utilised in this paper to identify several suspicious behaviours, like loitering, fainting, unauthorised access, etc. This method is based on how the various items move in relation to one another. First, a semantic method is used to define the various questionable activities. Next, background subtraction is used to detect objects. The discovered objects are then divided into living (human) and non-living categories (bag). These items must be tracked, which is accomplished via the correlation technique. Eventually, the events are categorised as normal or suspicious utilising the motion features & temporal information. This study employs a semantics-based approach to activity identification that relies on object tracking. The system's framework includes defining suspicious behaviour, backdrop subtraction, object detection, tracking, and activity classification. By applying the human interpretation of the activity, the semantic approach is used to define the suspicious activities. To identify the behaviour, the motion features between the two/different objects are retrieved. The semantic-based technique can be used to get around some of the drawbacks of machine learning, such as the lack of standard datasets and the classifier's lack of generalizability. The following is a list of the paper's contents: The research conducted by many researchers in this topic is examined in section two. The third section explains how the system functions and flows. The findings of the experiment are displayed in section four. It makes use of two objects' spatial relationships and motion characteristics. To find the specified activities of interest, the features are continuously compared to predetermined conditions. The method is straightforward for real-time performance and does not require training like machine learning-based approaches do. The future work of this project is to improve the system performances and prediction level.

2.2 Recognition of Suspicious Human Activities Using KLT and KalmanFilter For ATM Surveillance System

Author Name : Suvarna Nandyal and Sanjeev kumar Angadi

Year of Publish : 2021

In this project, the system which detects the suspicious and non-suspicious behaviors of human in ATM uses the Kanade Locus Thomasi algorithm which extracts features and object tracking to detect the suspicious activities this is a real time tracking system. Sensitive and public locations, including schools, colleges, jewellery stores, train stations, temples, banks, etc., can be watched using video surveillance to address suspicious activities. Following these people for a long period through such open spaces is tedious and time-consuming. The Automatic Teller Machine (ATM), which is under monitoring, is one such place. To protect the safety of ATMs, an intelligent monitoring system is presented that classifies real-time human behaviour into typical and unusual actions and can raise various levels of alarm. In order to detect and track suspicious or non-suspicious human behaviour for ATM video surveillance, the Kalman Filter and the Kanade-Lucas-Tomasi (KLT) Tracking Algorithm are proposed in this work. Results of experiments on a real-time ATM Surveillance database are carried out.

2.3 Alert Generation on Detection of Suspicious Activity Using TransferLearning

Author Name : Miwa Takai

Year of Publish : 2020

The proposed system provides additional feature to the CCTV camera's by providing a way to detect the suspicious activities like shop lifting, robbery by Using CNN model. The project's major objective is to use video surveillance to find suspicious behaviour and provide the consumers alert messages or notifications. This system uses videos from datasets as input and sends them to CNN model to determine whether or not our activity is suspicious. They intended to develop a tool for real-time identification of suspicious conduct by individuals in public settings. These can be applied to monitoring in locations where there is a possibility of theft or a shooting attack, such as malls, airports, train stations, etc. Its real-world uses range from gaming to healthcare to gesture recognition. The benefit of the suggested model is that it prevents crime before it occurs. Real-time CCTV footage is being monitored and examined. If the analysis's conclusion predicts an unfortunate incident will occur, the appropriate authority is instructed to take actions.

2.4 Suspicious Behavior Detection of People by Monitoring Camera

Author Name : Abouzar Ghasemi

Year of Publish : 2016

The system proposed a new approach based on the processing of the object trajectory for the detection of a suspicious behavior. The trajectory processing relies on the displacement vector of the interest object in. Fighting, running, leaving luggage and running, placing an odd packet somewhere unexpected like a dustbin, and leaving are some of the frequently seen suspicious activities in public places with a security component. They have concentrated on identifying suspicious activity and are looking for a way to employ computer vision techniques to discover a solution that can automatically detect suspicious activity. This topic has proven to be quite difficult, especially when it comes to real-time applications, because of the complex backgrounds, shifting lighting, and various distances between people and cameras. They used GMM to generate candidate regions with suspicious motion feature extraction from Optical Flow magnitude information; we refer to this approach as Suspicious Activity Region Detector (SARD). The robustness of our proposed framework over the state-of-the-arts in terms of both detection accuracy and processing speed, especially in congested settings, has been proved by experimental results on a number of benchmark datasets.

2.5 Suspicious Activity Detection from YOLOv3 Author

Name : Nipunjita Bordoloi, Anjan Kumar Talukdar

Year of Publish : 2021

YOLOv3 is used to identify a variety of suspicious activities, such as bag theft and lock-breaking. Both the system's processing speed and detection accuracy are very good. Only in a controlled environment does the current feature extraction technology provide correct findings. Regarding the processing time for a single image detection. This research developed a method for security system's suspicious action detection. In terms of processing time for a single image detection, we discovered that YOLOv3 performs better than Faster R-CNN. Only in a controlled environment does the current feature extraction technology provide correct findings. The outcomes can be enhanced by incorporating better feature extraction techniques. There were, however, some discrepancies between the test findings and the actual data due to the insufficient amount of training data. Consequently, expanding the training dataset to include suspicious films of various activities and resolutions is the future work to be done for development in order to get a better detection and make the model more useful. Moreover, more complex algorithms can be created for real-time applications.

2.6 Human Suspicious Activity Detection Using Ensemble Machine LearningTechniques

Author Name : Aqil Shamnath and Meena Belwal

Year of Publish : 2022

Public safety is an issue, particularly given our population growth. Effective crowd control is also required due to the growing population, as it is impossible to physically manage a sizable throng. Many solutions to this problem are being examined, and in this study, the solution has been developed using ensemble learning techniques that will catch suspicious activity. To identify, document, and warn the appropriate authorities to questionable activity, an automatic alert system has also been put up. Hence, the anomalous activity detection system will offer a fundamental monitoring system with an alarm, aiding in public safety while costing less and providing greater security. The decisions made by various models are combined in ensemble machine learning to enhance performance. We can identify suspicious behavior in public transportation, such as chain stealing and bag stealing, using what we've learned. This technique provides a detection accuracy of 88%.

2.7 Suspicious Activity Detection in Surveillance Footage Author

Name : Sathyajit Loganathan and Gayashan Kariyawasam

Year of Publish : 2019

It focuses on the analysis of two cases, including abandoned luggage and crimes involving guns that were captured on surveillance footage. In this method, computer vision and machine learning pipelines are used to detect abandoned luggage in videos and handguns in photos. In order to identify probable gun-related crimes and abandoned luggage scenarios in surveillance footage, this study presents a deep neural network model that can recognize firearms in photos as well as a machine learning and computer vision pipeline that can identify abandoned luggage. Our research advises experimenting with various designs and comparing them to maximize speedier predictions for weapons as future work for this topic. Due to concerns about time and resource constraints, we were only able to bring up the research to the point that is discussed in this report, allowing for future research on how to enhance the detection of guns in real time. It may also be a good idea to conduct research on adding elements other than surveillance footage to improve real-time detections. Further research on this topic can be done to advance the field since the approach for finding abandoned luggage in this study does not handle problems such item detection in unexpected changes of illumination.

2.8 Deep Learning Approach for Suspicious Activity Detection fromSurveillance Video

Author Name : Amrutha C. V and Joytsna Amudha J

Year of Publish : 2020

The separation of various human suspicious behavior from the camera footage is made possible by the integration of machine learning and deep learning. There are two sections to the complete structure. In the first section, video frame features are calculated, and in second, a classifier predicts whether a given class is suspicious or normal based. Video surveillance is crucial in today's society. When machine learning, deep learning, and artificial intelligence were introduced to the system, the technologies had already evolved too far. Several methods are in place that help to distinguish distinct suspicious activities from the live tracking of footages using the combinations mentioned above. Human behaviour is the most erratic, and it can be quite challenging to determine whether it is normal or suspicious. In an academic setting, a deep learning approach is utilised to identify suspect or regular behaviour. If suspicious activity is predicted, the approach alerts the appropriate authority. Consecutive frames taken from the video are frequently used for monitoring. There are two sections to the complete structure. In the first section, video frame features are calculated, and in the second section, a classifier predicts whether a given class is suspicious or normal based on the features that have been gathered.

2.9 Abnormal behavior recognition for intelligent video surveillance systems

Author Name : Amira Ben Mabrouk and Ezzeddine Zagrouba

Year of Publish : 2017

The behaviour representation and the behaviour modelling, the two key components of a video surveillance system, are the focus of this research. The review covers feature extraction and description techniques for behaviour representation. Frameworks for behaviour modelling and classification approaches are also offered. In addition, presented datasets and measures for performance evaluation. Lastly, real-world applications of current video surveillance systems are detailed. This review has looked at the behaviour representation and behaviour modelling layers of a video surveillance system. They started by reviewing the most widely utilised techniques for features extraction and description. Next they gave a thorough discussion of several frameworks and categorization techniques for behaviour modelling.

2.10 Understanding User Behavior through Action Sequences: From theUsual to the Unusual

Author Name : Phong H. Nguyen, Cagatay Turkay

Year of Publish : 2019

Action sequences, which are timestamped, labelled representations of atomic user activities, are increasingly important data assets for examining and tracking user behaviour in digital systems. Although the research of such sequences is crucial for cyber security applications, current techniques fall short of offering a whole understanding because of the complex semantic and temporal properties of these data. In order to assist a user-involved, multifaceted decision-making process during the detection and study of "strange" activity sequences, this paper provides a visual analytics technique. They first reported the results of the task analysis and domain characterization. Then they described the components of the multi-level analysis approach that comprises of constraint-based sequential pattern mining and semantic distance based clustering, and multi-scalar visualizations of users and their sequences. Lastly, they provide a case study that incorporates decisions that need to be made effectively by a group of domain experts to show how their methodology can be used. They have given the findings and methodologies that are applicable to other applications where the analysis of such sequences is of interest, despite the fact that their solution in this case is strongly informed by a user-centered, domain-focused design process.

2.11 Human behavior recognition method based on double-branch deep convolution neural network

Author Name : Zhou Zhigang, Duan Guangxue, Lei Huan

Year of Publish : 2018

This research provides a method of human behavior recognition based on double-branch deep convolution neural network. To begin with, the characteristics of the input image are extracted, and the feature maps are then fed into a double-branch deep convolution neural network to produce information on the joints of the human body and the connections between those joints.

2.12 Suspicious human activity recognition: a review Author

Name : Rajesh Kumar Tripathi and Anand Singh Jalal

Year of Publish : 2017

The most recent state-of-the-art that demonstrates the overall development of the detection of suspicious activity from surveillance films over the past ten years is described in this study. They have provided a succinct introduction to the detection of suspicious human activity, along with its problems and difficulties. The paper consists of six abnormal activities such as abandoned object detection, theft detection, fall detection, accidents and illegal parking detection on road, violence activity detection, and fire detection. The detection of suspicious human activity in surveillance footage is a current field of study for image processing and computer vision. In order to prevent terrorism, theft, accidents and illegal parking, vandalism, fighting, chain snatching, crime, and other suspicious activities, human activity can be observed in sensitive and public places such as bus stations, railway stations, airports, banks, shopping malls, schools and colleges, parking lots, and roads through visual surveillance. As it is exceedingly challenging to constantly monitor public spaces, it is necessary to install sophisticated video surveillance that can track people's movements in real-time, classify them as routine or exceptional, and provide alerts. There have been a lot of papers in the last ten years about using visual surveillance to spot unusual activity. It used to identify human activity in surveillance films by feature extraction, object identification and tracking or non-tracking techniques.

2.13 CCTV Face Detection Criminals and Tracking System Using DataAnalysis Algorithm

Author Name : Patiyuth Pramkeaw, Pearlrada Ngamrungsiri, Mahasak Ketcham

Year of Publish : 2019

The algorithm uses technology to reduce the amount of effort spent looking for criminals and identifying suspicious. Study used a face detection approach to examine CCTV footage. As a comparison, the results of single face detection and group face detection were obtained. The program's accuracy was 91%.

2.14 A Visual Analytics Approach for User Behavior Understanding throughAction Sequence Analysis

Author Name : P. H. Nguyen and C. Turkay

Year of Publish : 2017

In this project, to provide a rich understanding of user behavior. Then, in order to take the first steps towards a thorough understanding of user behavior, they provide the elements of a visual analytics technique that is a novel combination of "action space" analysis, pattern mining, and the interactive visual analysis of several sequences. Using the analysis of user activity sequences, they have detailed the first stages of a visual analytics approach that intends to enable a deep understanding of user behaviour. They noticed that the complex nature of action sequences necessitates a detailed examination of sessions from many angles and through comparisons, rendering visualisation a feasible strategy.

2.15 Automated Real-Time Detection of Potentially Suspicious Behavior inPublic Transport Areas

Author Name : Martin D. Levine

Year of Publish : 2013

This system proposes the investigation of the behavior modelling and representation—the two key components of a video surveillance system. This review covers feature extraction and description techniques for behavior representation. Frameworks for behavior modelling and classification approaches are also offered. In this paper, the authors describe a framework that analyses unprocessed video data from a fixed colour camera deployed at a specific position to infer activities in real time. Using a real-time blob matching technique, the proposed system first detects and tracks people and luggage in the scene to collect 3-D object-level information. By using object and inter object motion cues, behaviours and events are semantically detected based on the temporal characteristics of these blobs. To illustrate the capabilities of this strategy, a variety of behaviours that are pertinent to security in transit hubs have been chosen. They include things that have been abandoned or stolen, as well as fighting, fainting, and loitering. The framework for displaying the power of surveillance in the current system, which defeats the need for labor-intensive human surveillance to find out what's going on. Their method translates the photos into scene semantic segmentation, which aids in the detection of motion features that have been previously captured. They have included a function that immediately sends an alarm if any variation in the collected scenes occurs, and it also stores each caught scene for future use as a reference.

CHAPTER 3
SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In the existing models, it is crucial to learn suspicious human poses in order to recognize suspicious human activities. It has to do with identifying human bodily parts and perhaps monitoring their motion. Nevertheless, due to their low resolution and noisy depth information, these sensors are only suitable for indoor application, making it challenging to infer human movement from depth photos. As a result, they are not a good choice for spotting suspicious activities. There are certain models like OpenPose, which gives out the key point coordinates of the people int the video which is not enough to predict the activity is Violence. The drawbacks from the existing models are low resolution data, less no of perspectives in data, detects the activity in footage (not real time), image frame comparison.

3.1.1 Disadvantages of Existing System

While object detection is a valuable and widely used technology, there are some potential disadvantages to consider:

- Complexity: Item detection uses intricate algorithm and consumes a lot of processing power. This means that both its implementation and operation can be time consuming and costly.
- System requirements: The hardware level needs to be higher than usual in order for the system to operate well.
- Data requitements: It takes a lot of labelled data to train an object detection model. The cost of gathering and annotating this can make it challenging to obtain.

3.2 PROPOSED SYSTEM

In our project, we have decided to overcome the drawbacks of the existing systems by using Convolutional Neural Network Model. By using sequential model, we are able to extract the features from the Data frames like,

- High resolution data
- various variety of illumination in the data
- large number of perspectives in the data

The above data constraints are used to extract the feature and train the model using deep learning model and detect the violence and then alerting the respective authority regarding the activity.

3.2.1Advantages of Proposed System

- Elevated consciousness: The system is capable of detecting violations and alerting the appropriate user to take appropriate action.
- Weapons detection: Identification of weapons usage in sensitive places is possible.
- Increased efficiency: Businesses and organizations can increase efficiency and cut costs by automation object detection tasks.

3.3 FEASIBILITY STUDY

Security guards may start to make mistakes and fail to notice critical moments when monitoring is necessary when they are forced to focus on numerous security monitors at once. Hours of monitoring may pass with no unusual activity, yet when a break-in takes place, the security officer may be caught off guard. He may have to choose between apprehending the criminal and letting him get away if he loses a few moments of video. More security officers must be hired, which is expensive and does not necessarily fix the issue. A mechanism mustbe created to help security personnel perform their duties more effectively.

3.4 HARDWARE ENVIRONMENT

- RAM: 8GB
- Processor: AMD Ryzen 5k series with NVIDIA-GTX 4GB graphics.
- Cameras

3.5 SOFTWARE ENVIRONMENT

- Python
- Jupyter Notebook (Worksapce)
- Keras-Tensorflow (Deep learning Library)
- Pyttsx3 – for Voice Generation.
- OpenCV– library for Capturing video.

CHAPTER 4
SYSTEM DESIGN

4. SYSTEM DESIGN

4.1 DATA FLOW DIAGRAM

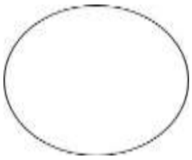
4.1.1 Data Flow Diagram Level 0

To illustrate the movement of information throughout a procedure or system, one might use a Data-Flow Diagram (DFD). A data-flow diagram does not show any loops or decision-making processes because information only flows in one direction. You can use a flowchart to show the steps necessary to complete a certain data-driven task. Data-flow graphs can be represented in a variety of ways. Each data flow must have a process that serves as either the information exchange's source or its goal. The activity diagram is frequently used instead of a data-flow diagram by UML users. Site-oriented data-flow plans are a subset of the larger category of data-flow plans. The semantics of data memory are represented by the locations in the network, therefore identical nodes in a data-flow diagram and a Petri net can be thought of as inverted equivalents. DFM (structured data modelling) contains terminators, flows, storage, and processes.

Data Flow Diagram Symbols

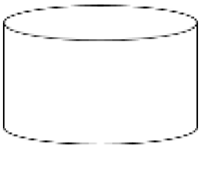
Process

A process is one that takes in data as input and returns results as output.



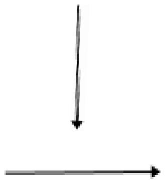
Data Store

In the context of a computer system, the term "data stores" is used to describe the various memory regions where data can be found. In other cases, "files" might stand in for data.



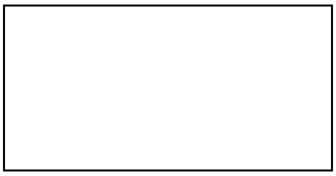
Data Flow

Data flows are the pathways that information takes to get from one place to another. Please describe the nature of the data being conveyed by each arrow.



External Entity

In this context, "external entity" refers to anything outside the system with which the system has some kind of interaction. These are the starting and finishing positions for inputs and outputs, respectively.



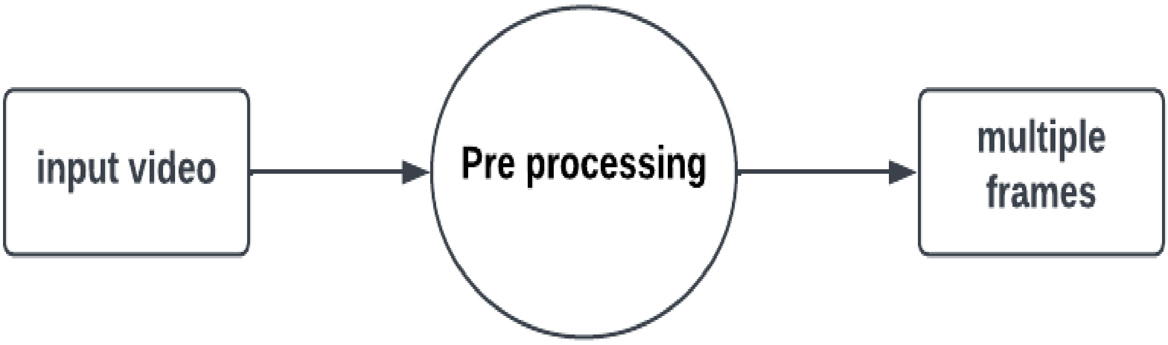


Fig 4.1.1 – Data Flow Diagram Level 0

4.1.2 Data Flow Diagram Level 1

A level DFD displays the entire system as a single process. The system's assembly process is documented here in detail, including all intermediate steps. These two diagrams, along with 2- level data flow diagrams, make up the "basic system model."

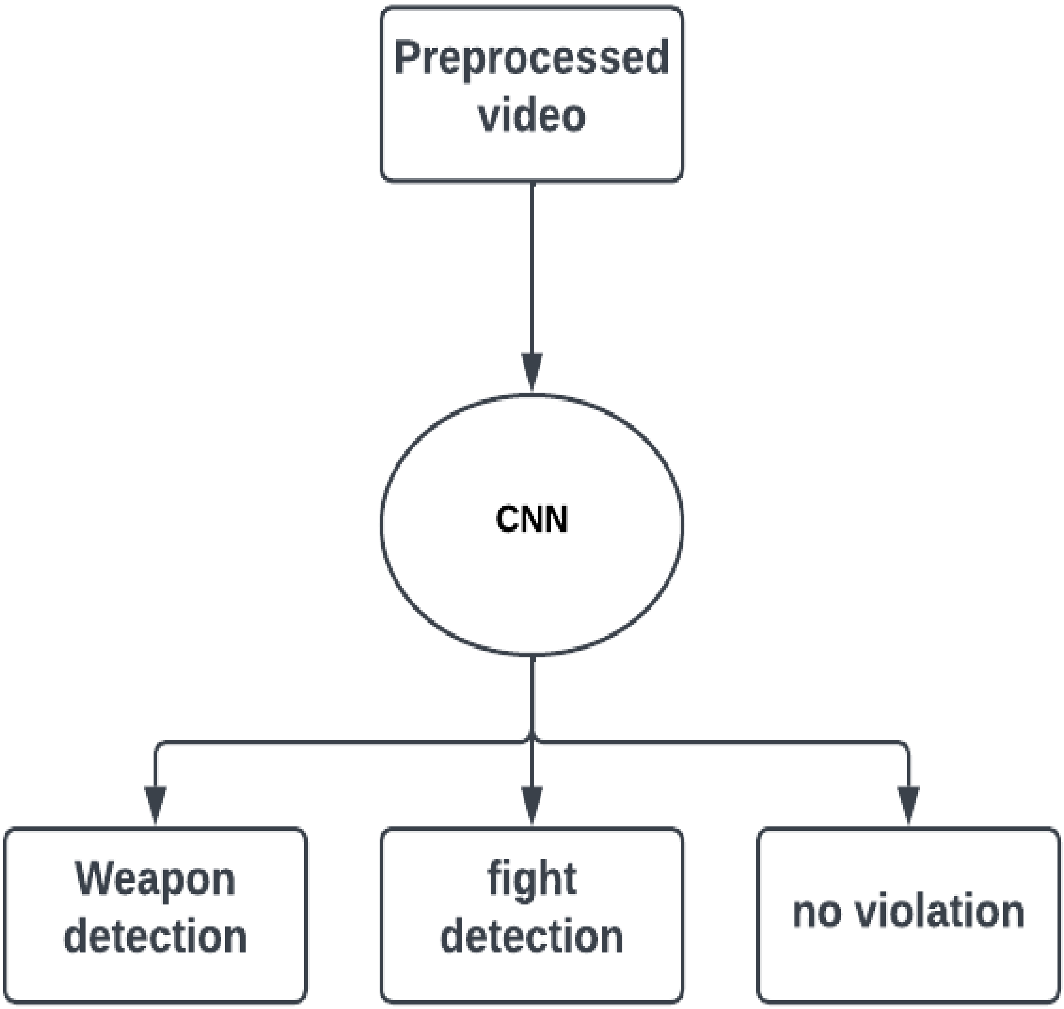


Fig 4.1.2 – Data Flow Diagram Level 1

4.1.3 Data Flow Diagram Level 2

2-level DFD goes one process deeper into parts of 1-level DFD. It can be used to project or record the specific/necessary detail about the system's functioning.

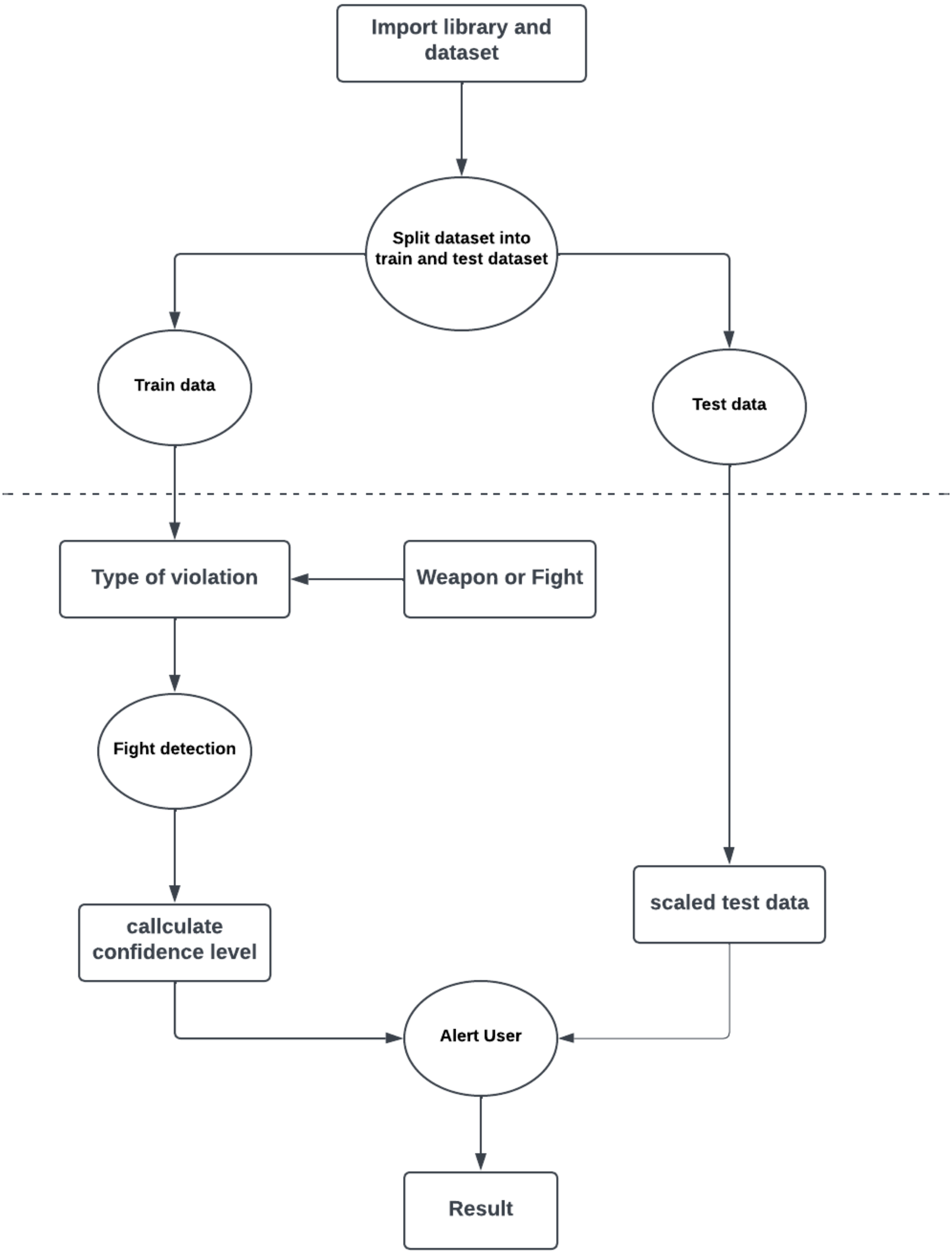


Fig 4.1.3 – Data Flow Diagram Level 2

4.2 UML DIAGRAMS

A UML diagram is a diagram based on the UML (Unified Modelling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artefacts or classes, in order to better understand, alter, maintain, or document information about the system. It is based on diagrammatic representations of software components.

Some UML diagrams are:

- Use case diagram
- Class diagram
- Activity diagram
- Sequence diagram
- State chart diagram
- Component diagram
- Deployment diagram

4.2.1 Use-Case Diagram

The possible interactions between the user, the dataset, and the algorithm are often depicted ina use case diagram. It's created at the start of the procedure.

- Actors: Actors are external entities that interact with the system. They can be human users, other systems, or devices.
- Use Cases: Use cases are the specific functions or tasks that the system can perform. Each use case represents a specific interaction between an actor and the system.
- Relationships: Relationships are used to indicate how the actors and use cases are related to each other. The two main relationships in a use case diagram are "uses" and "extends". "Uses" relationship indicates that an actor uses a specific use case, while "extends" relationship indicates that a use case extends or adds functionality to another use case.
- System Boundary: The system boundary is a box that contains all the actors and use cases in the system. It represents the physical or logical boundary of the system being modeled.
- Use case diagrams are useful for identifying the functional requirements of a system, and for communicating these requirements to stakeholders. They can be used in the requirements gathering phase of software development, as well as in the design and testing phases.

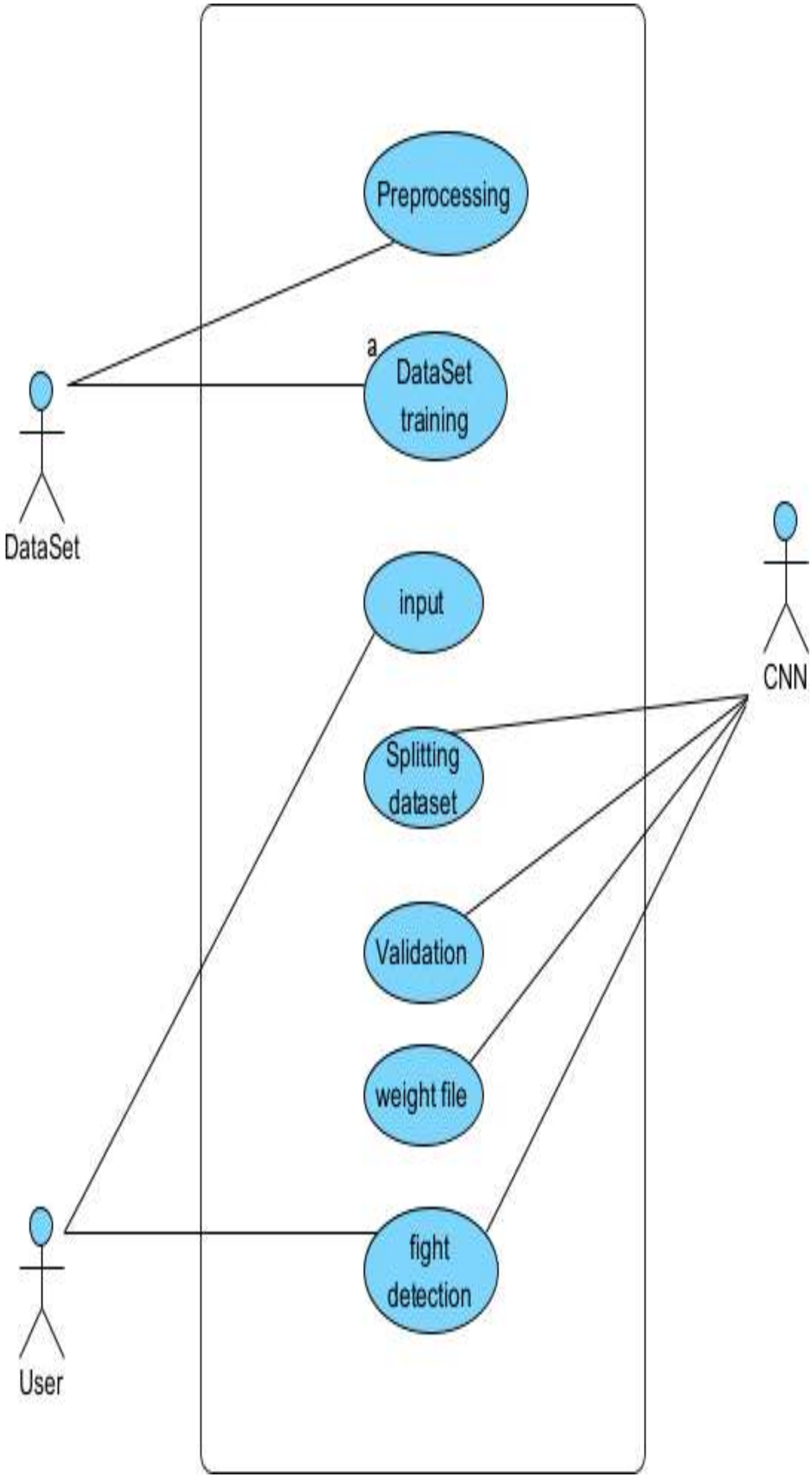


Fig 4.2.1 – Use-Case Diagram

4.2.2 Sequence Diagram

These are another type of interaction-based diagram used to display the workings of the system. They record the conditions under which objects and processes cooperate. A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram

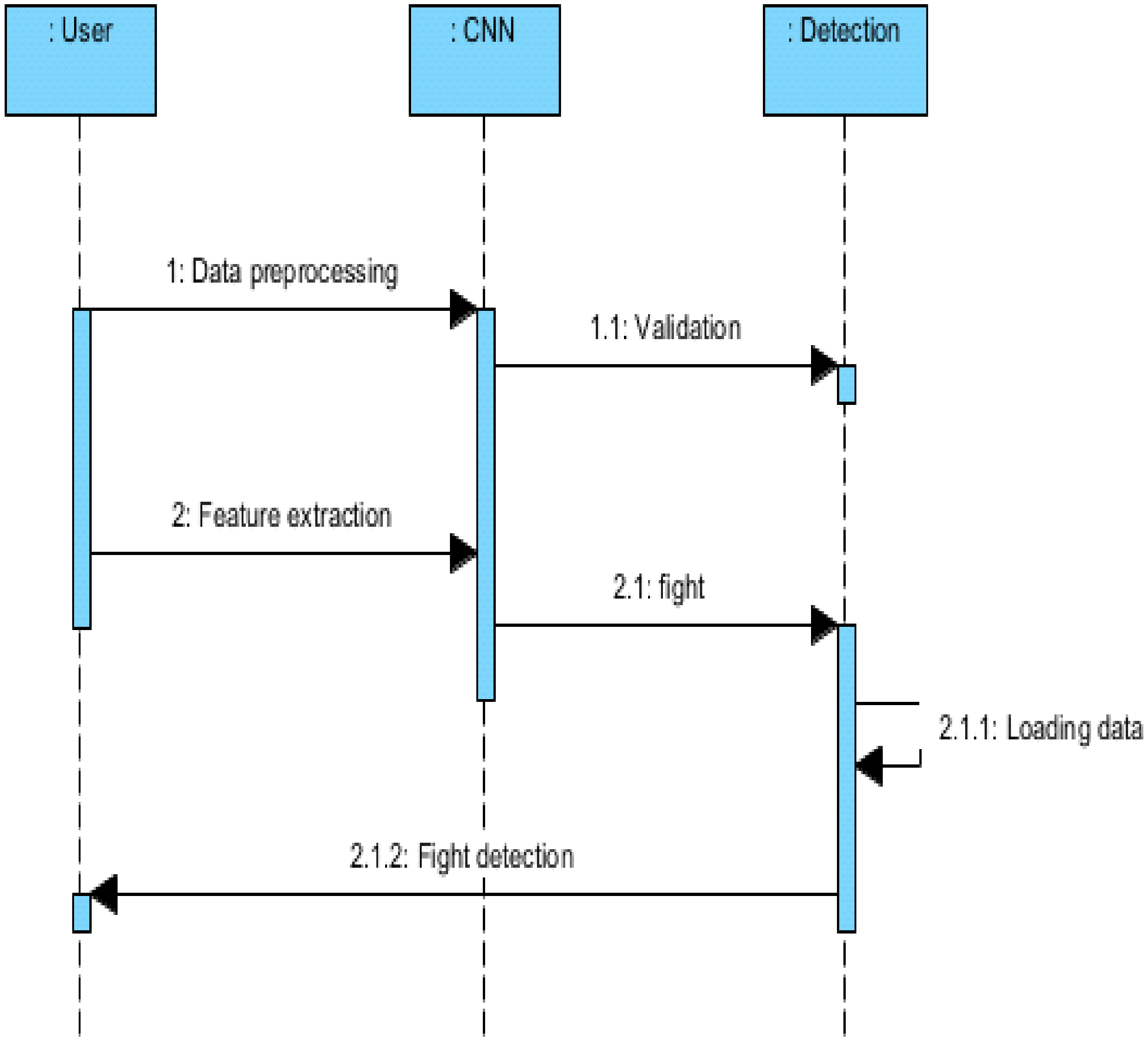


Fig 4.2.2 – Sequence Diagram

4.2.3 Activity Diagram

A clear and straightforward explanation of each entity currently incorporated into the system is given in this diagram. The picture illustrates the relationships between the various options and activities. You could say that the entire procedure and the manner in which it was completed paint an image. The functional links between different entities are depicted in the image below.

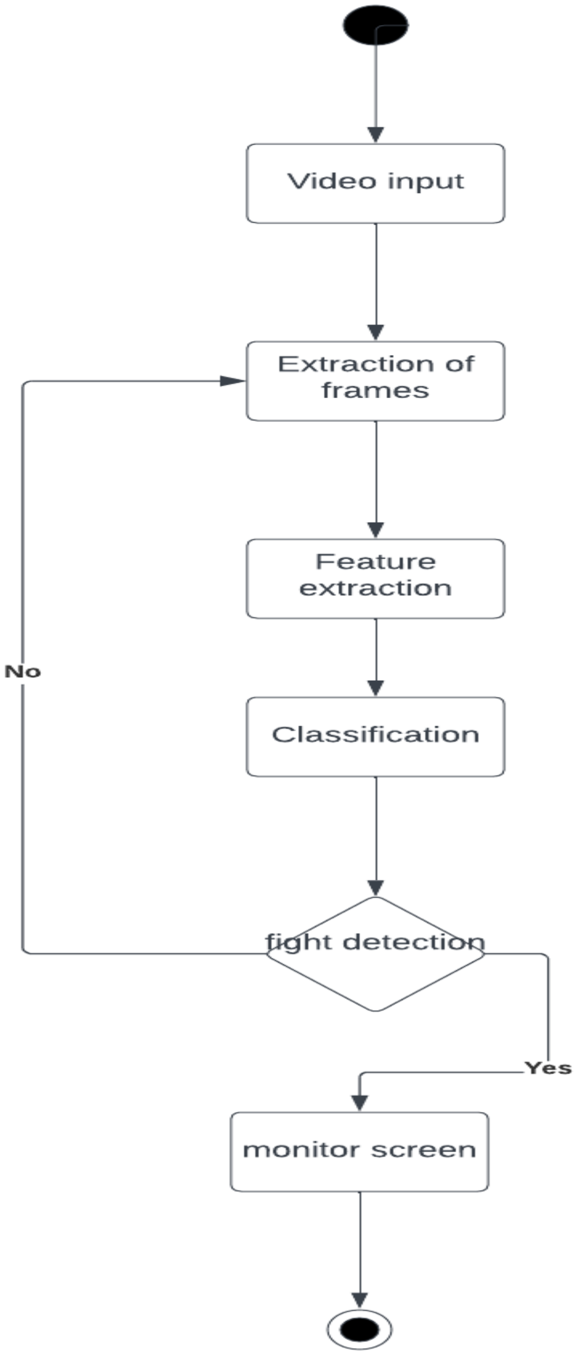


Fig 4.2.3 – Activity Diagram

4.2.4 Class Diagram

In essence, this is a "context diagram," another name for a contextual diagram. It simply stands for the very highest point, the 0 Level, of the procedure. As a whole, the system is shown as asingle process, and the connection to externalities is shown in an abstract manner.

- A + indicates a publicly accessible characteristic or action.
- A - a privately accessible one.
- A # a protected one.
- A - denotes private attributes or operations.

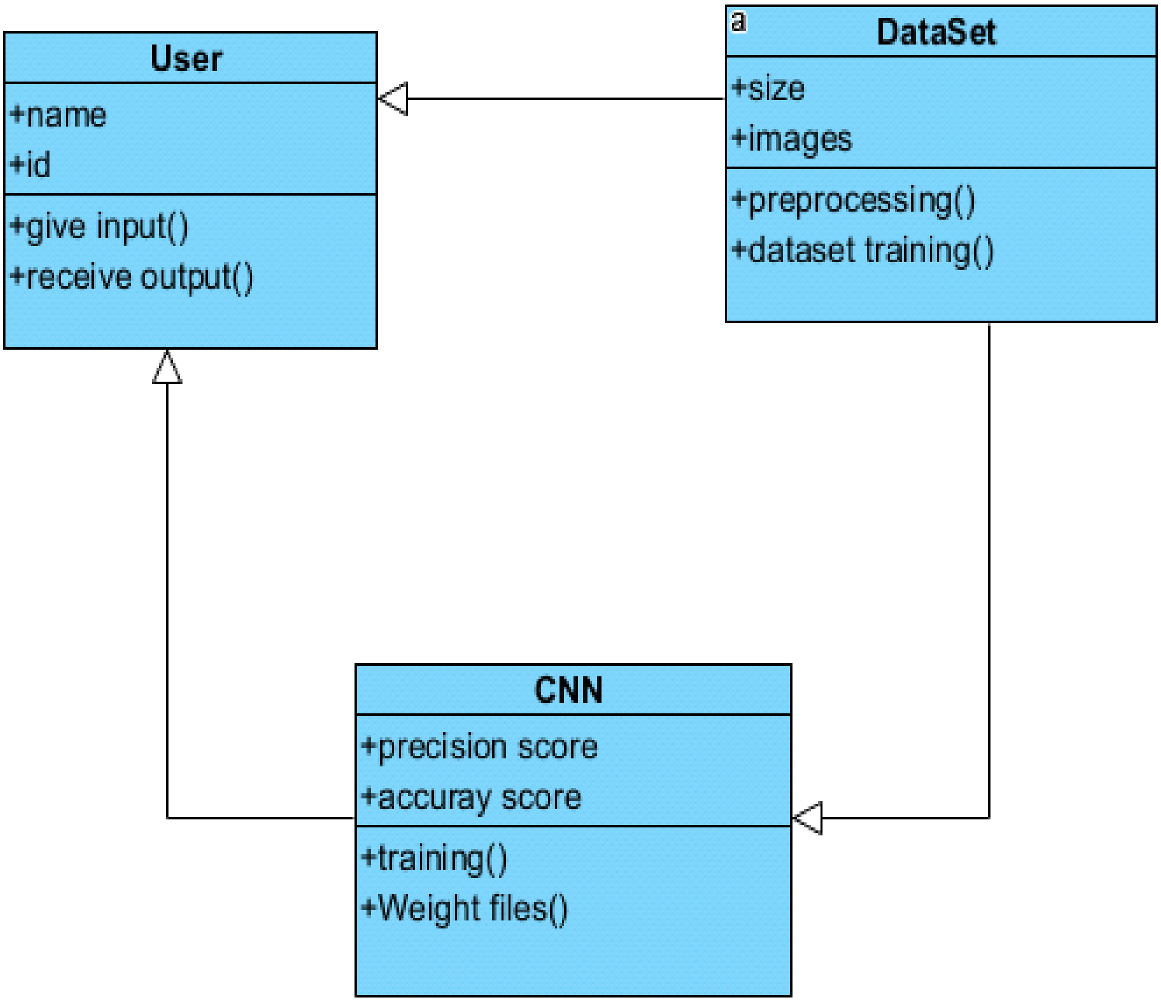


Fig 4.2.4 – Class Diagram

4.2.5 State Chart Diagram

State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State chart diagram is to model lifetime of an object from creation to termination. State chart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

Following are the main purposes of using State chart diagrams

- To model the dynamic aspect of a system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object.

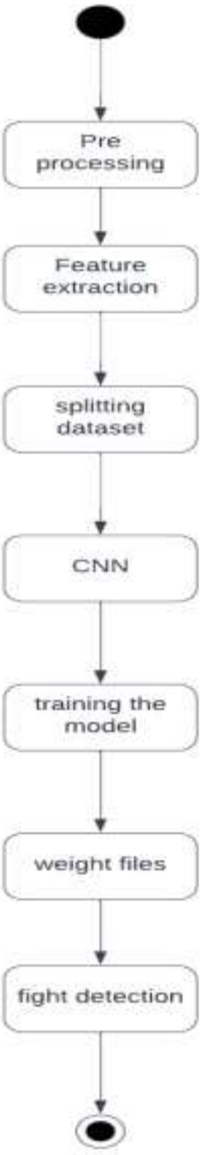


Fig 4.2.5 – State Chart Diagram

4.2.6 Component Diagram

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node. It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behavior is explained.

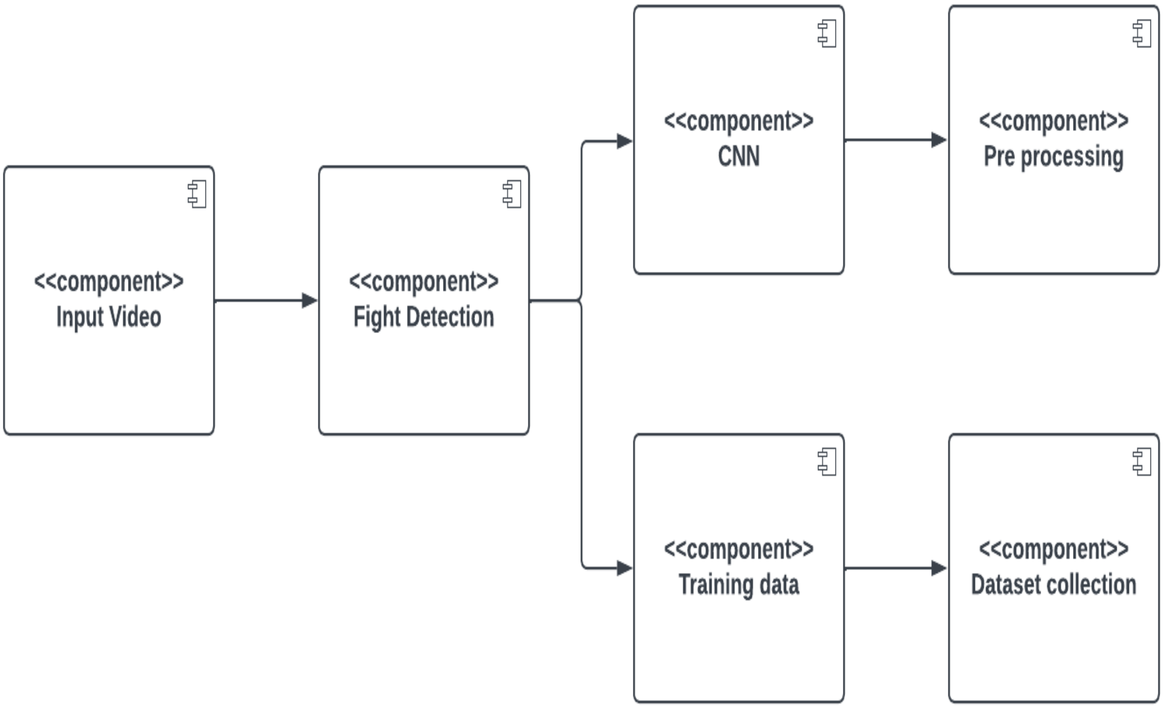

<<component>> Input Video
<<component>> Fight Detection
<<component>> CNN
<<component>> Pre processing
<<component>> Training data
<<component>> Dataset collection

Fig 4.2.6 - Component Diagram

4.2.7 Deployment Diagram

The deployment diagram visualizes the physical hardware on which the software will be deployed. It portrays the static deployment view of a system. It involves the nodes and their relationships. It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths.

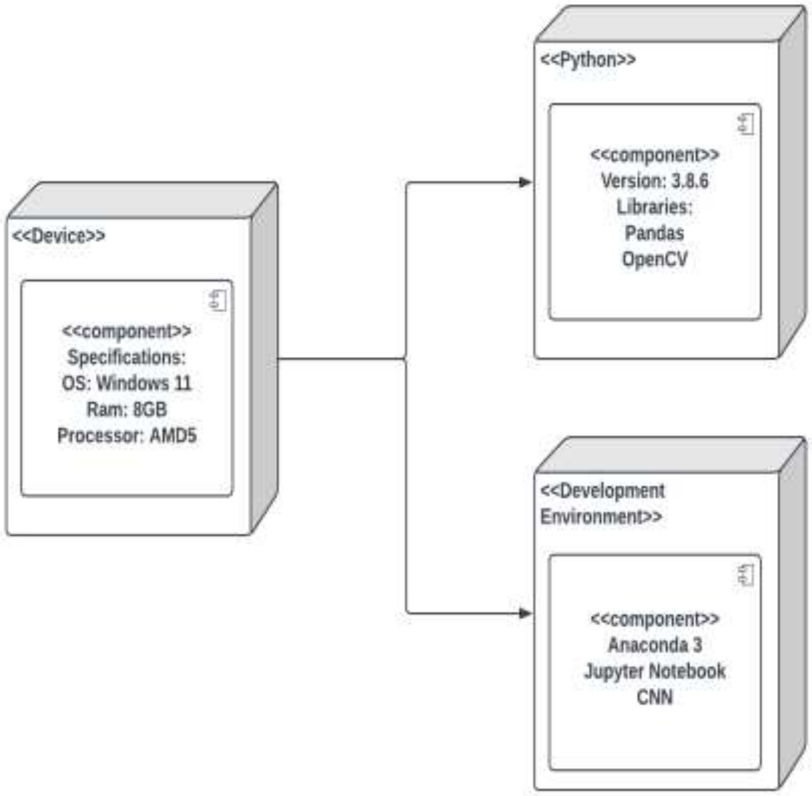


Fig 4.2.7- Deployment Diagram

CHAPTER 5
SYSTEM ARCHITECTURE

+

5. SYSTEM ARCHITECTURE

5.1 SYSTEM ARCHITECTURE

All of the entities that are currently incorporated into the system are described in this image in a clear and succinct manner. The figure illustrates the relationships between the various activities and decisions. The entire procedure and the way it was handled might be described as a picture. The functional links between distinct entities are depicted in the image below.

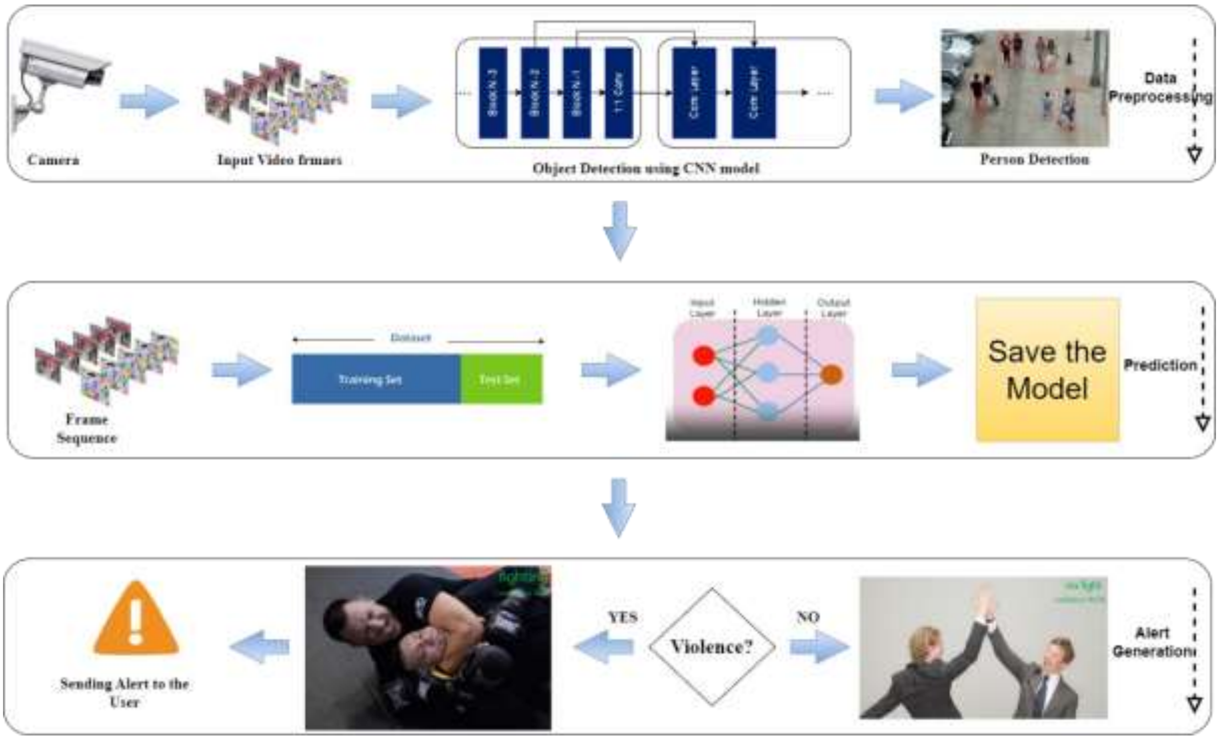


Fig 5.1 Architecture diagram

In fig 5.1 the system architecture for anonymous activity detection can be divided into several components, including:

➢ Input Data: A camera mounted at an angle can provide the system with the input data it needs. The camera records pictures or videos of the humans and their activities.

➢ Pre-processing: Pre-processing is used to improve the picture quality once it has been captured. In doing so, it might be necessary to reduce noise, tweak the luminance and contrast, and fix distortion.

➢ Data Classification: Based on the categories of suspicious human activity they contain, surveillance video data sets are gathered and categorized. Datasets on violence, robbery, and fire have been obtained. They include real-world examples of shady behaviour from various places.

- ➢ Frame Extraction: The frames of the videos are separated using a Python script. The appropriate frames that accurately and clearly depict the action of anonymous are manually chosen for use. The videos from which the extracted frames were taken feature a variety of settings, events, people, and situations. Every action, about 250–300 frames are chosen.
- ➢ Labelling: The quality of the data labelling has a significant impact on how accurate our application is. XML Python package is used to label training and test datasets. By carefully choosing the region of label, the name of the suggested suspicious human activity is carefully labelled on each frame. After labelling, the frames are stored as.xml files.
- ➢ Data Conversion and CNN: For picture categorization, we employed a straightforward CNN model, with datasets in.jpg format as its input.The.xml files are transformed to CSV files using Python scripts, and then their files are translated to TF Records format.
- ➢ Training and Testing Sets: By dividing the total amount of data into a predetermined ratio for each activity, the selected frames are further divided into training dataset and testing dataset.
- ➢ Person detection: It is used to detect whether the person performs suspicious activity or not. Object detection and facial recognition are two examples of machine-learning methods that can be used for this.
- ➢ Result: After training is complete, the application recognises suspicious human activity by displaying coloured, labelled boxes near the incident for video input. When using an image as an input, the GUI displays the results and a percentage of the output.

5.2 ALGORITHM

5.2.1 Convolutional Neural Network

A Convolutional Neural Network (CNN) is a type of Deep Learning neural network architecture commonly used in Computer Vision. Computer vision is a field of Artificial Intelligence that enables a computer to understand and interpret the image or visual data. The data is fed into the model and output from each layer is obtained from the above step is called feedforward, we then calculate the error using an error function, some common error functions are cross-entropy, square loss error, etc. The error function measures how well the network is performing. After that, we backpropagate into the model by calculating the derivatives. This step is called Backpropagation which basically is used to minimize the loss.

5.2.2 CNN Architecture

Convolutional Neural Network consists of multiple layers like the input layer, Convolutional layer, Pooling layer, and fully connected layers.

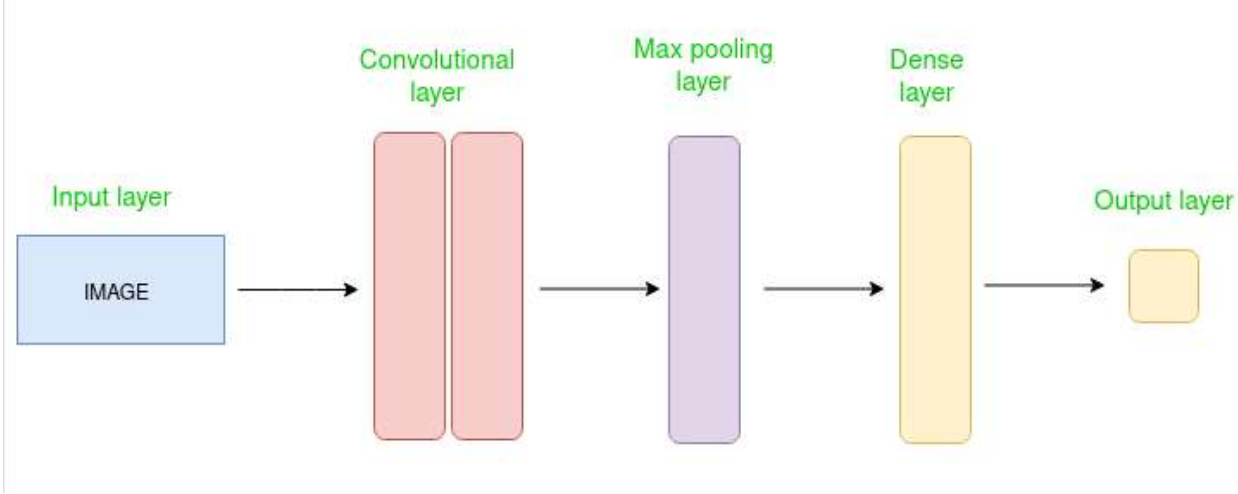


Fig 5.2.2 CNN Architecture

In fig 5.2.2 the Convolutional layer applies filters to the input image to extract features, the Pooling layer down samples the image to reduce computation, and the fully connected layer makes the final prediction. The network learns the optimal filters through backpropagation and gradient descent.

5.2.3 Working of CNN

Convolution Neural Networks or coverts are neural networks that share their parameters. Imagine you have an image. It can be represented as a cuboid having its length, width (dimension of the image), and height (i.e the channel as images generally have red, green, and blue channels).

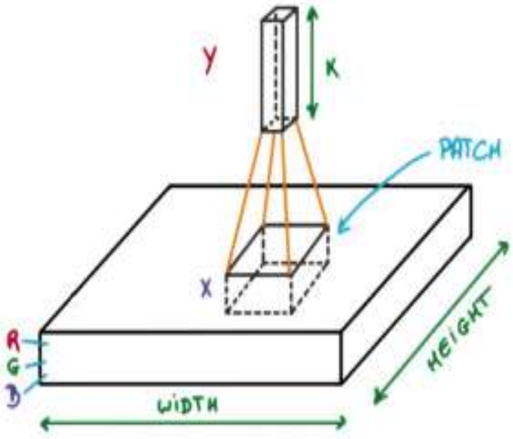


Fig 5.2.2 Working of CNN

In fig 5.2.3 now imagine taking a small patch of this image and running a small neural network, called a filter or kernel on it, with say, K outputs and representing them vertically. Now slide that neural network across the whole image, as a result, we will get another image with different widths, heights, and depths. Instead of just R, G, and B channels now we have more channels but lesser width and height. This operation is called Convolution. If the patch size is the same as that of the image it will be a regular neural network. Because of this small patch, we have fewer weights.

CHAPTER 6

SYSTEM IMPLEMENTATION

6. SYSTEM IMPLEMENTATION

6.1 MODULE DESCRIPTION

The system is made up of three main parts:

- Dataset Collection and Preprocessing
- Model Training
- Prediction of Output

6.1.1 Module 1: Data Collection and Preprocessing

The construction of a reliable and effective violence detection system depends heavily on the data collecting and preprocessing module for CNN (Convolutional Neural Network)- based violence detection. Data collection and preparation are the module's two main processes. Building a machine learning model always starts with gathering data, and CNN violence detection is no different. The module needs gathering pertinent photos and videos of collisions from a variety of sources, including dashcams, CCTV cameras, and other sources. To guarantee the model's correctness and dependability, the dataset needs to be substantial, diverse, and balanced. Data labelling, which is the next phase and essential for CNN-based Violence detection,is required. To detect a fight and any important objects in the picture, such as a gun or knife, the technique entails scaling the photographs and videos. Given that the accuracyof the output from the model depends on the quality of the input data, the resizing andnoise removal procedure calls for an understanding of OpenCV and a strong eye for detail.

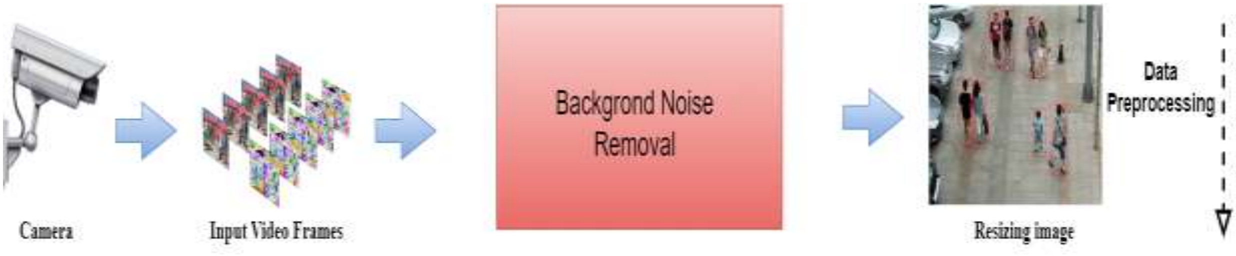


Fig 6.1.1 – Data Collection and Preprocessing

In fig 6.1.1 the data can be used to train the model for detecting violence once it has been preprocessed and transformed into the necessary format. Data collection and preprocessing are crucial steps in creating a reliable and effective system for detecting violent crimes because the quality of the training data directly affects the model's performance. In conclusion, the data gathering and preprocessing module for CNN-based violence detection is an essential step in creating a reliable and effective system. To gather pertinent information, appropriately label it, and preprocess it into the format necessary for the model training, the module demands careful organisation, attention to detail, and domain experience.

6.1.2 Module 2: Model Training

The creation of a reliable and effective violence detection system depends heavily on the model training module for CNN (Convolutional Neural Network)-based violence detection. The module comprises training the model using the preprocessed data, which is a part of machine learning. A deep neural network that has been trained using a sizable dataset of images and videos makesup the CNN model. To determine the bounding boxes and class probabilities for each object inan image or video, the model employs a convolutional neural network. The preprocessed data is fed into the CNN model during the model training phase, and it is trained to identify and categorize items in the image or video. Using gradient descent, the model's parameters are optimized during the training phase to reduce the discrepancy betweenexpected and actual results.

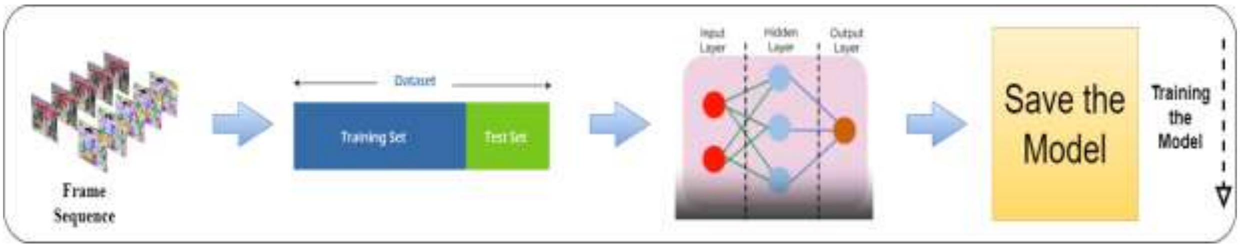


Fig 6.1.2 – Model Training

In fig 6.1.2 the training data should be varied, extensive, and balanced to guarantee the model's correctness and dependability. A wide range of violence scenarios, lighting circumstances, and other pertinent elements that may affect the model's accuracy should be included in the training data. After the model has been trained, transfer learning strategies can be used to improve it. A limited collection of photos and videos particular to the intended application is used to fine- tune the model, increasing its efficacy and accuracy. In conclusion, the CNN model training module for violence detection is essential for creating a reliable and effective system for detecting violence. In this module, the model is trained on preprocessed data, its parameters are optimized, and it is then fine-tuned using transfer learningstrategies. The quality and diversity of the training data determine how accurate and reliable the model is, therefore the module calls for careful planning, attention to detail, and competence in machine learning and computer vision.

6.1.3 Module 3: Prediction of Output

The final stage in creating a reliable and effective violence detection system is the prediction of output module for violence detection using CNN. The module entails applying the trained CNN model to forecast the likelihood of violence and any pertinent scene elements, such as a knife or gun. Preprocessed data, including photos and videos, must be input into the trained model in order to forecast the output module. A confidence score and class probability for the violence in thescene are produced by the model after it has analysed the input data. The class probabilities show the possibility that an action falls under a particular class, such a fight or not. The CNN model's output can be seen in real time on a monitor or presented to the user as an alert. Depending on the needs of the application, the alert may take the form of a notification, sound, or alarm. The result can be post-processed using a variety of methods, including thresholding, to increase the precision and dependability of the predictions. Setting a minimal threshold for the class probabilities reduces false positives and increases the precision of the model. For real-time violence detection applications, when quick responses are necessary to prevent or lessen the intensity of violence, the prediction of output module is vital.

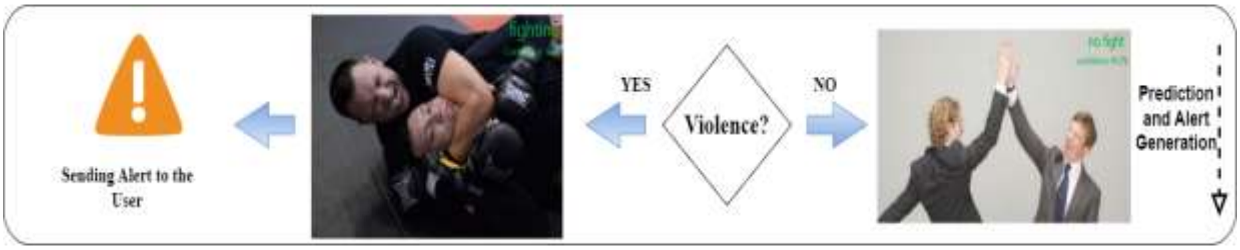


Fig 6.1.3 – Prediction of Output

In fig 6.1.3 the correctness of the trained model, the efficacy of the post-processing approaches, and the calibre of the preprocessed data all affect how accurate and efficient the module is. In conclusion, the CNN output module prediction for violence detection is essential for creating a reliable and effective violence detection system. The module entails applying the trained model to forecast the occurrence of violence and any pertinent scene elements, visualising the results in real-time, and post-processing the results to increase accuracy and dependability.

CHAPTER 7
TESTING

7. TESTING

7.1 WHITE BOX TESTING

The box testing approach of software testing consists of black box testing and white box testing. We are discussing here white box testing which also known as glass box is testing, structural testing, clear box testing, open box testing and transparent box testing. It tests internal coding and infrastructure of a software focus on checking of predefined inputs against expectedand desired outputs. It is based on inner workings of an application and revolves around internalstructure testing. In this type of testing programming skills are required to design test cases. The primary goal of white box testing is to focus on the flow of inputs and outputs through thesoftware and strengthening the security of the software.

The term 'white box' is used because of the internal perspective of the system. The clearbox or white box or transparent box name denote the ability to see through the software's outershell into its inner workings. Developers do white box testing. In this, the developer will test every line of the code of the program. The developers perform the White-box testing and then send the application orthe software to the testing team, where they will perform the black box testing and verify the application along with the requirements and identify the bugs and sends it to the developer. The developer fixes the bugs and does one round of white box testing and sends it to the testing team. Here, fixing the bugs implies that the bug is deleted, and the particular featureis working fine on the application.

The white box testing contains various tests, which are as follows:

- Path testing
- Loop testing
- Condition testing
- Testing based on the memory perspective
- Test performance of the program

7.2 BLACK BOX TESTING

Black box testing is a technique of software testing which examines the functionality of software without peering into its internal structure or coding. The primary source of black box testing is a specification of requirements that is stated by the customer. In this method, tester selects a function and gives input value to examine its functionality, and checks whether the function is giving expected output or not. If the function produces correct output, then it ispassed in testing, otherwise failed. The test team reports the result to the development team andthen tests the next function. After completing testing of all functions if there are severe problems, then it is given back to the development team for correction.

The test procedure of black box testing is a kind of process in which the tester has specific knowledge about the software's work, and it develops test cases to check the accuracy of the software's functionality. It does not require programming knowledge of the software. All test cases are designedby considering the input and output of a particular function. A tester knows about the definite output of a particular input, but not about how the result is arising. There are various techniques used in black box testing for testing like decision table technique, boundary value analysis technique, state transition, All-pair testing, cause-effect graph technique, equivalence partitioning technique, error guessing technique, use case technique and user story technique.

7.3 TEST CASES

TEST REPORT : 01

USE CASE **:** Upload Input Video

TEST CASE ID	**ACTION TO BE PERFORMED**	**EXPECTED RESULT**	**ACTUAL RESULT**	**PASS/FAIL**
1	Upload the Video frames of .mp4 as input	Uploaded	Uploaded	Pass
2	Upload the Video frames of .avi as input	File format must be changed and uploaded	Supported	Pass

Table -7.3.1 Test Case for Input Video

TEST REPORT : 02

USECASE : Search and Detect Violence Behaviour

TEST CASE ID	ACTION TO BE PERFORMED	EXPECTED RESULT	ACTUAL RESULT	PASS/FAIL
1	Detect the Violence Behavior in Day time	Detected Successfully	Detected Successfully	Pass
2	Detect the Violence Behavior in Night time with Light	Detected Successfully	Detected Successfully	Pass
3	Detect the Violence Behavior in Night time without light	Detected Successfully	Detected	Pass
4	Detect the Violence Behavior in Crowded area	Detected Successfully	Detected Successfully	Pass

Table-7.3.2 Search and Detect the Violence Behavior

TEST REPORT : 03

USECASE : Generating Alarm to the User

TEST CASE ID	ACTION TO BE PERFORMED	EXPECTED RESULT	ACTUAL RESULT	PASS/FAIL
1	Alert the Violence Behavior in Day time	Alerted Successfully	Alerted Successfully	Pass
2	Alert the Violence Behavior in Night time with Light	Alerted Successfully	Alerted Successfully	Pass
3	Alert the Violence Behavior in Night time without light	Alerted Successfully	Not Alerted	Pass
4	Alert the Violence Behavior in Crowded area	Alerted Successfully	Alerted Successfully	Pass

Table-7.3.3 Generating Alarm to the User

CHAPTER 8
CONCLUSION AND FUTURE ENHANCEMENT

8. CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION

In the modern world, practically everyone is aware of the value of CCTV footage, yet in the majority of cases, these footages are only used for investigation after a crime or incident has occurred. According to the project, to identify suspicious activity in surveillance footage, convolutional neural networks for feature extraction and discriminative deep belief networks for action categorization should be used. The suggested approach provides better categorizationthan past attempts by utilizing a deep-learning-based model. Then, using CNN, we extracted characteristics from the background and foreground of the movie after dividing it into frame segments. A trained DDBN then receives the output and organizes the recognized actions as normal or suspicious. More precision and fewer false positives are promised by the deep learning model. The benefit of the suggested paradigm is that it prevents crime from developing into a serious scenario. Real-time CCTV footage is being monitored and examined. If the analysis's findings suggest an unfortunate incident is likely to occur, the appropriate authorityis instructed to take actions. Thus, it is possible to stop this.

8.2 FUTURE ENHANCEMENT

In the future, the model can be improved by developing more interactive and data visualization tools that help future investigators quickly identify suspicious activities. The focus on the user interface experience for the end users can be enhanced. The introduction and utilization of blockchain technology to the model can produce secure and safe data related to abnormal or suspicious activities which can provide greater transparency and accountability, as well as improve data integrity and privacy.

APPENDICES

A.1 Coding

Model_train.ipynb

```
# Importing the tensorflow library for model.
     import tensorflow as tf


# Optionally, the first layer can receive an `input_shape` argument:
    model = tf.keras.Sequential()
    model.add(tf.keras.layers.Dense(8, input_shape=(16,)))


# Afterwards, we do automatic shape inference:
    model.add(tf.keras.layers.Dense(4))


# This is identical to the following:
    model = tf.keras.Sequential()
    model.add(tf.keras.Input(shape=(16,)))
    model.add(tf.keras.layers.Dense(8))


# Note that you can also omit the `input_shape` argument.
# In that case the model doesn't have any weights until the first call#
To a training/evaluation method (since it isn't yet built):
    model = tf.keras.Sequential()
    model.add(tf.keras.layers.Dense(8))
    model.add(tf.keras.layers.Dense(4))
# model.weights not created yet


# Whereas if you specify the input shape, the model gets built#
continuously as you are adding layers:
    model = tf.keras.Sequential()
    model.add(tf.keras.layers.Dense(8, input_shape=(16,)))
    model.add(tf.keras.layers.Dense(4))
    len(model.weights)
# Returns "4"
# When using the delayed-build pattern (no input shape specified), you can
# choose to manually build your model by calling
#  `build(batch_input_shape)`: model
    = tf.keras.Sequential()
    model.add(tf.keras.layers.Dense(8))
    model.add(tf.keras.layers.Dense(4))
    model.build((None, 16))
    len(model.weights)
# Returns "4"

# Note that when using the delayed-build pattern (no input shape specified)# the
model gets built the first time you call `fit`, `eval`, or `predict`,
# or the first time you call the model on some input data.
    model = tf.keras.Sequential()
    model.add(tf.keras.layers.Dense(8))
    model.add(tf.keras.layers.Dense(1))
    model.compile(optimizer='sgd', loss='mse')


     # This builds the model for the first time:
    model.fit(x, y, batch_size=32, epochs=10)

# Saving the Model:
    model.save(“keras_Model.h5”)
```

Prediction.ipynb

```
#Importing and Loading the Model
   from keras.models import load_model

# TensorFlow is required for Keras to work
   import cv2 # Install opencv-python
   import numpy as np
   import time
   import pyttsx3
```

```
# Create TTS engine
   engine = pyttsx3.init()

# Set voice properties
   voices = engine.getProperty('voices')
   engine.setProperty('voice', voices[1].id)
   engine.setProperty('rate', 170)

# Disable scientific notation for clarity
   np.set_printoptions(suppress=True)

# Load the model
   model = load_model("keras_Model.h5", compile=False)
   model1 = load_model("keras_Model1.h5",compile=False)



# CAMERA can be 0 or 1 based on default camera of your computer
   camera = cv2.VideoCapture(0)
   start_time = time.time()
   fight_duration = 0
   fcount =0
   fighting = False

   while cv2.waitKey(1):
# Grab the webcamera's image.
       ret, image = camera.read()
       img = image
        if not ret:
           cv2.waitKey()
           print(fight_duration)
           break
```

```
# Show the image in a window
    cv2.imshow("Webcam Image", image)

# Resize the raw image into (224-height,224-width) pixels
    image = cv2.resize(image, (224, 224), interpolation=cv2.INTER_AREA)


# Make the image a numpy array and reshape it to the models input shape.
    image = np.asarray(image, dtype=np.float32).reshape(1, 224, 224, 3)

# Normalize the image array
    image = (image / 127.5) - 1

# Predicts the model
    prediction = model.predict(image)
    index = np.argmax(prediction)
    prediction1 = model1.predict(image)
    index1 = np.argmax(prediction1)

#class_name = class_names[index]
#confidence_score = prediction[0][index]
    if index == 0 :
        if not fighting:
            start_time = time.time()
        fighting = True
        fcount +=1
        fight_duration += time.time() - start_time
    else:
        fcount = 0
        fight_duration = 0
        fighting = False
    if fcount> 10:
        str1 = "camera 1"
```

```
# Convert text to speech
           text = "Emergency !!!, Fighting at {}".format(str1)
           engine.say(text)
           engine.runAndWait()
          fcount = 0

    # Listen to the |keyboard for presses.
       keyboard_input = cv2.waitKey(1)
       if keyboard_input == 27:
           break

  camera.release()
  cv2.destroyAllWindows()
```

A.2 SAMPLE SCREENS

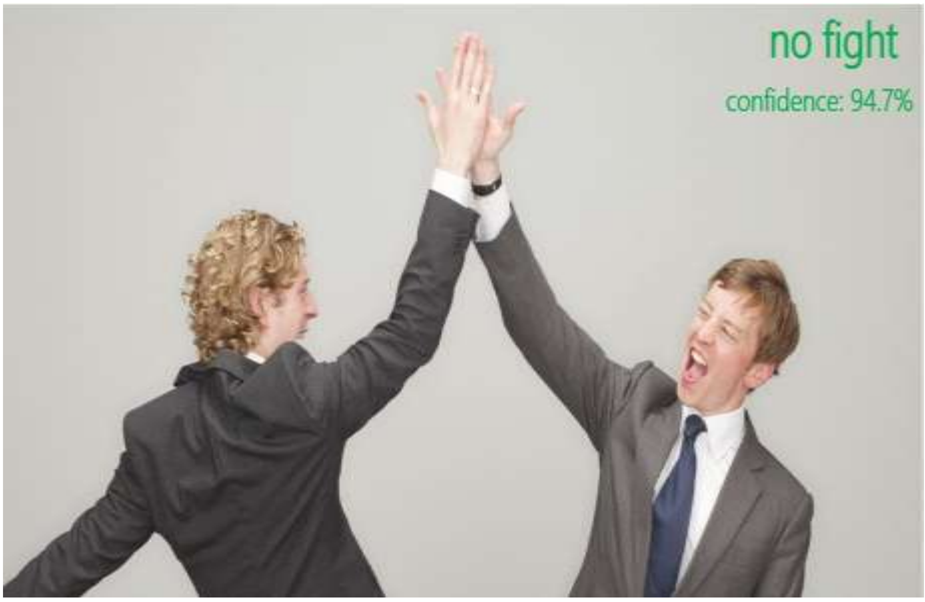


Fig A.2.1- High five and No fight

In fig A.2.1 two persons are giving high five and the model detects it as no fight with confidence of 94.7%.

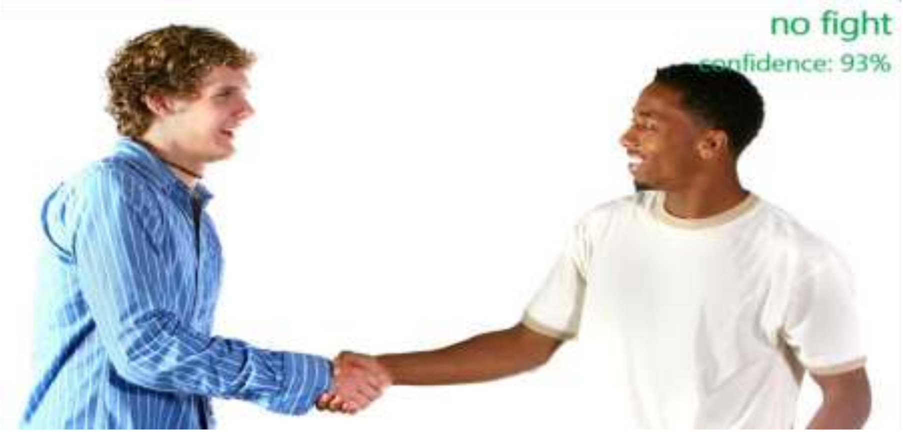


Fig A.2.2- Hand Shake and No fight

In fig A.2.2 two persons are giving Hand shake to each other and the model detects it as no fight with confidence of 93%.

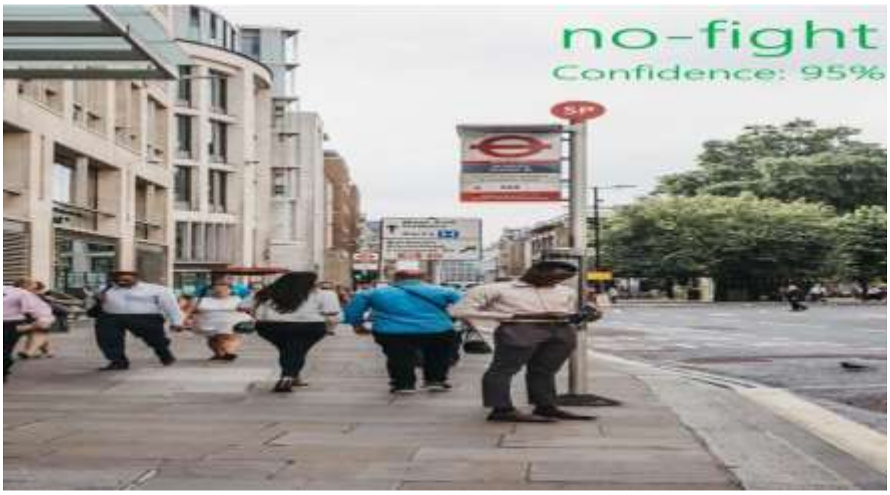


Fig A.2.3-Standing in Crowded area and no fight

In fig A.2.3, All the people are doing their own activity and no violence is done there. Hence the machine identifies it as No Fight with a confidence score of 95%.

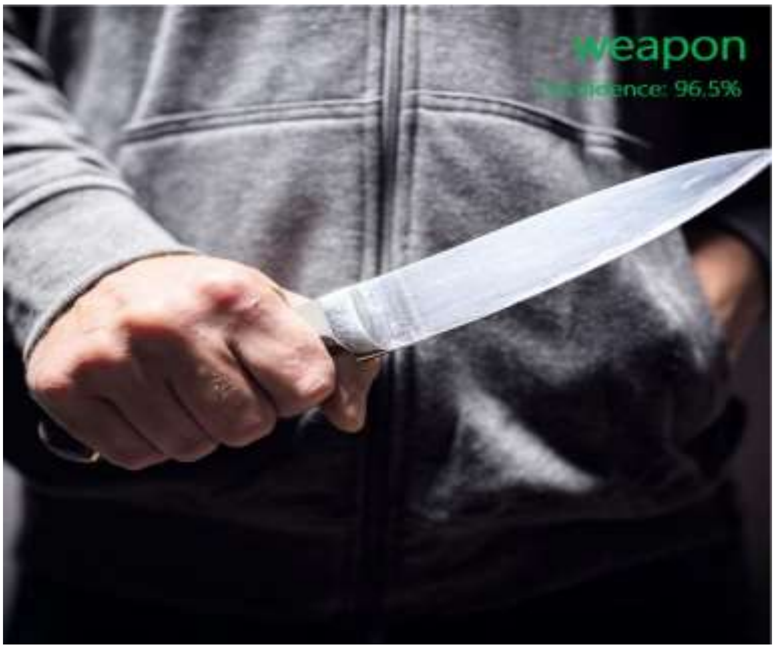


Fig A.2.4- Man with weapon

In fig A.2.4, A man was standing with a knife in his hand, so a violence is going tobe done. Hence the machine identifies it as weapon with a confidence score of 96.5%.

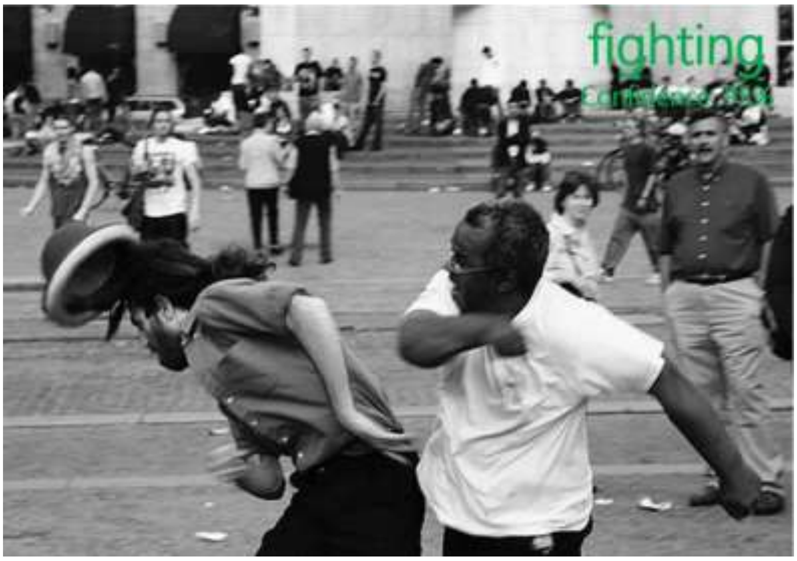


Fig A.2.5- Two persons fighting

In fig A.2.5, two men are fighting in a public place. so the machine identifies it as a Fight with a confidence score of 95%.

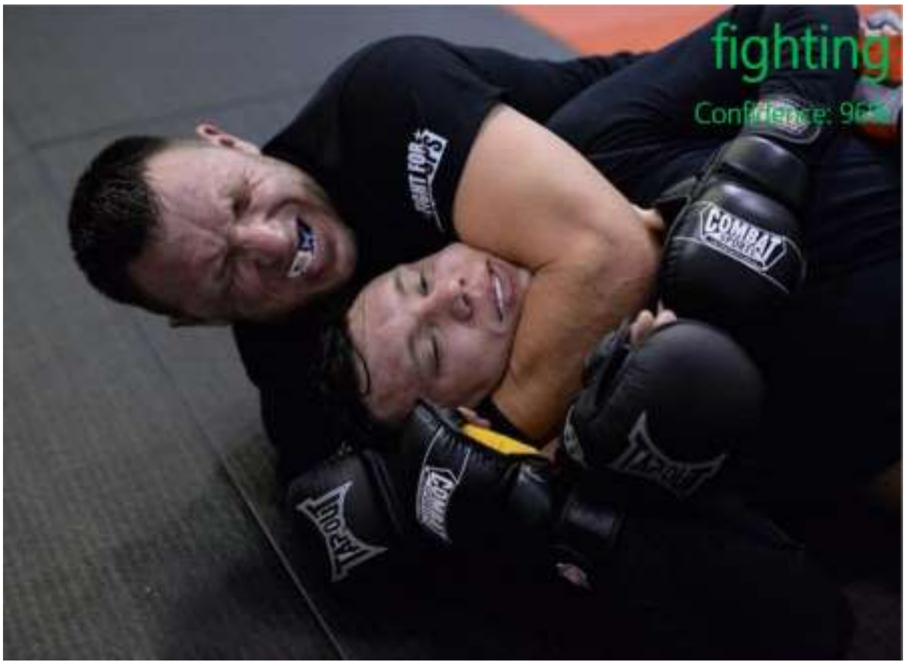


Fig A.2.6- A person locking another one neck

In fig A.2.6, two men are fighting (which means they are playing combat sports). Hence the machine identifies it as Fight with a confidence score of 96%.

REFERENCES

REFERENCES

[1] Ms. U. M. Kamthe and Dr. C. G. Patil “Suspicious Activity Recognition in Video Surveillance System”, Fourth International Conference on Computing Communication Control and Automation, Pune, India, Aug. 2018, pp. 1–6.

[2] Suvarna Nandyal and Sanjeev kumar Angadi “Recognition of Suspicious Human Activities Using KLT and Kalman Filter For ATM Surveillance System”, in International Conference on Innovative Practices in Technology and Management, Noida, India, Feb. 2021, pp. 174–179.

[3] Miwa Takai “Alert Generation on Detection of Suspicious Activity Using Transfer Learning”, in Second World Congress on Nature and Biologically Inspired Computing, Kitakyushu, Japan, Dec. 2010, pp. 298–304.

[4] Abouzar Ghasemi “Suspicious Behavior Detection of People by Monitoring Camera”, 2016. International Journal of Computer Trends and Technology (IJCTT) – volume 29 Number 1.

[5] Nipunjita Bordoloi, Anjan Kumar Talukdar “Suspicious Activity Detection from YOLOv3”, 2021.IEEE 10.1109/INDICON49873.2020.9342230.

[6] Aqil Shamnath and Meena Belwal “Human Suspicious Activity Detection Using Ensemble Machine Learning Techniques”, 2022. IEEE 10.1109/CONIT55038.2022.9848183.

[7] Sathyajit Loganathan and Gayashan Kariyawasam “Suspicious Activity Detection in Surveillance Footage”, in International Conference on Electrical and Computing Technologies and Applications, Ras Al Khaimah, United Arab Emirates, Nov. 2019, pp. 1–4.

[8] Amrutha C. V and Joytsna Amudha J “Deep Learning Approach for Suspicious Activity Detection from Surveillance Video”, 2020.IEEE ICIMIA48430.2020.9074920.

[9] Phong H. Nguyen, Cagatay Turkay “Understanding User Behavior through Action Sequences: From the Usual to the Unusual”, 2019. IEEE TVCG.2018.2859969.

[10] Rajesh Kumar Tripathi and Anand Singh Jalal “Suspicious human activity recognition: a review”, Artificial Intelligence Review, vol. 50, no. 2, pp. 283–339, Aug. 2018.

[11] P. H. Nguyen and C. Turkay “A Visual Analytics Approach for User Behavior Understanding through Action Sequence Analysis”, 2017.

[12] Amira Ben Mabrouk and Ezzeddine Zagrouba “Abnormal behavior recognition for intelligent video surveillance systems”, 2017. 10.1016/j.eswa.2017.09.029.

[13] Zhou Zhigang, Duan Guangxue, Lei Huan “Human behavior recognition method based on double-branch deep convolution neural network”, 2018.

[14] Patiyuth Pramkeaw, Pearlrada Ngamrungsiri, Mahasak Ketcham “CCTV Face Detection Criminals and Tracking System Using Data Analysis Algorithm”, 2019.

[15] Martin D. Levine “Automated Real-Time Detection of Potentially Suspicious Behavior in Public Transport Areas”, 2013. IEEE TITS.2012.2228640.