Weapon Detection in Real-Time CCTV Videos Using Deep Learning

A PROJECT REPORT

Submitted in partial fulfillment of requirements to

R.V.R & J.C COLLEGE OF ENGINEERING

For the award of the degree

B.Tech in CSE

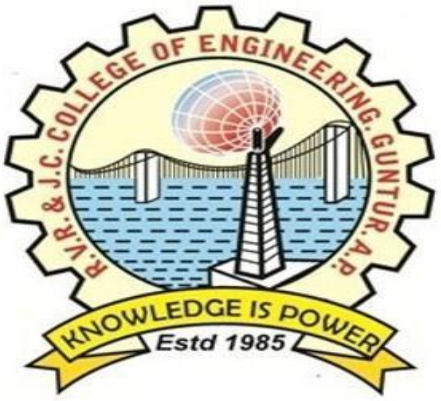
by

Batch No: 4

B.NANDINI (Y19CS003)

CH.SWETHA(Y19CS024)

CH.ROOPA DEVI(Y19CS023)



December 2022

R.V.R. & J.C. COLLEGE OF ENGINEERING (AUTONOMOUS)

(NAAC ‘ A+ ’ Grade)

(Approved by AICTE, Affiliated to Acharya Nagarjuna University)

Chandramoulipuram: Chowdavaram,
Guntur - 522019, Andhra Pradesh, India

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that project work titled **"WEAPON DETECTION IN REALTIME CCTV VIDEOS USING DEEP LEARNING"** is the work done and submitted by **B.NANDINI (Y19CS003), CH.SWETHA (Y19CS024) and CH.ROOPA DEVI (Y19CS023),** in partial fulfilment of the requirements to the **Project-II (CS-452),** during the academic year 2022-2023.

Dr. R. Lakshmi Tulasi	**Dr. B. Vara Prasad Rao**	**Dr.M.Sreelatha**
Project Guide	Project In-charge	Prof. and HOD, CSE

ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without a proper suggestions, guidance and environment. Combination of these three factors acts like backbone to our Project "**WEAPON DETECTION IN REAL-TIME CCTV VIDEOS USING DEEPLEARNING**".

We are very glad to express our special thanks to **Dr. R. Lakshmi Tulasi,** our Project Guide, who has inspired us to select this topic, and also for her valuable advices in preparing this Project work.

We are very thankful to **Dr. B. Vara Prasad Rao Sir**, Project In-charge for the Project who extended his encouragement and support to carry out this Project successfully.

We are greatly indebted to **Dr. M. Sreelatha**, Professor and Head of the Department of Computer Science and Engineering, for her valuable suggestion, encouragement and support during the course period.

We are very much thankful to **Dr. K. Ravindra**, Principal of **R.V.R. &J.C. College of Engineering**, Guntur for providing the supportive Environment.

Finally we submit our reserve thanks to lab staff in Department of Computer Science and Engineering and to all our friends for their cooperation during this project work preparation.

Batch No: 4

B.NANDINI (Y19CS003)

CH.SWETHA(Y19CS024)

CH.ROOPADEVI(Y19CS023)

ABSTRACT

Security and safety is a big concern for today's modern world. For a country to be economically strong, it must ensure a safe and secure environment for investors and tourists. Having said that, Closed Circuit Television (CCTV) cameras are being used for surveillance and to monitor activities i.e. robberies but these cameras still require human supervision and intervention. We need a system that can automatically detect these illegal activities. Despite state-of-the-art deep learning algorithms, fast processing hardware, and advanced CCTV cameras, weapon detection in real-time is still a serious challenge. Observing angle differences, occlusions by the carrier of the firearm and persons around it further enhances the difficulty of the challenge. This work focuses on providing a secure place using CCTV footage as a source to detect harmful weapons by applying the state of the art open-source deep learning algorithms. We have implemented binary classification assuming pistol class as the reference class and relevant confusion objects inclusion concept is introduced to reduce false positives and false negatives. No standard dataset was available for real-time scenario so we made our own dataset by making weapon photos from our own camera, manually collected images from internet, extracted data from YouTube CCTV videos, through GitHub repositories. Some of the algorithms used are VGG16, Inception-V3, Inception-ResnetV2, SSDMobileNetV1, Faster-RCNN Inception-ResnetV2 (FRIRv2), YOLOv3, and YOLOv4. Precision and recall count the most rather than accuracy when object detection is performed so these entire algorithms were tested in terms of them. Yolov4 stands out best amongst all other algorithms and gave a F1-score of 91% along with a mean average precision of 91.73% higher than previously achieved.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of Abbreviations

CCTV	-	Closed Circuit Television
IMFDB	-	Internet Movies Firearms Database
CWD	-	concealed weapon detection
CNN	-	convolutional neural network
COCO	-	common objects in context
CCD	-	Charge-Coupled Device
ROI	-	region of interest
SURF	-	Speed up Robust Features
HOG	-	Histogram of oriented Gradient
ResNet	-	Residual Neural Network
YOLO	-	You Only Look Once
RCNN	-	Region-based CNNs
GPU	-	Graphic Processing Unit
ML	-	Machine Learning
AP	-	average precision
mAP	-	mean average precision

1. INTRODUCTION

1.1 Background

The crime rate across the globe has increased mainly because of the frequent use of handheld weapons during violent activity. For a country to progress, the law-and-order situation must be in control. Whether we want to attract investors for investment or to generate revenue with the tourism industry, all these needs is a peaceful and safe environment. The crime ratio because of guns is very critical in numerous parts of the world. It includes mainly those countries in which it is legal to keep a firearm. The world is a global village now and what we speak or write has an impact on the people. Even if the news they heard is crafted having no truth but as it gets viral in a few hours because of the media and especially social media, the damage will be done. People now have more depression and have less control over their anger, and hate speeches can get those people to lose their minds. People can be brainwashed and psychological studies show that if a person has a weapon in this situation, he may lose his senses and commit a violent activity.

CCTV cameras play an important role to overcome this problem and are considered to be one of the most important requirements for the security aspect. [3]. CCTVs are installed in every public place today and are mainly used for providing safety, crime investigation, and other security measures for detection. CCTV footage is the most important evidence in courts. After a crime is committed, law enforcement agencies arrive at the scene and take the recording of footage with them [4].

The methodology adopted in this work features the state of art deep learning, especially the convolutional neural networks due to their exceptional performance in this field. [40]. The aforementioned techniques are used for both the classification as well as localizing the specific object in a frame so both the object classification and detection algorithms were used and because our object is small with other object in background so after experimentation we found the best algorithm for our case. Sliding window/classification and region proposal/object detection algorithms were used, and these techniques will be discussed later in this section.

1.2 Problem Statement

Violence committed with guns puts significant impact on public, health, psychological, and economic cost. Many people die each year from gun-related violence. Psychological trauma is

frequent among children who are exposed to high levels of violence in their communities or through the media. Children exposed to gun-related violence, whether they are victims, perpetrators, or witnesses, can experience negative psychological effects over the short and long terms. Number of studies show that handheld gun is the primary weapon used for various crimes like break-in, robbery, shoplifting, and rape. These crimes can be reduced by identifying the disruptive behaviour at early stage and monitoring the suspicious activities carefully so that law enforcement agencies can further take immediate action .

CCTV is one of the possible operational requirements to be considered in a wider security and safety aspects. CCTV cameras are being installed almost every where in public places for providing security. Main use of CCTV surveillance are for providing security, deterrence, crime investigation and reduction in insurance costs. Deterrence effects of CCTV cameras varies for different crime categories and even from time to time. There are many places where the crime rate caused by weapons is very high, especially in places where they are allowed. The early detection of potentially violent situations is of paramount importance for citizens security. One way to prevent these situations is by detecting the presence of dangerous objects such as handguns and knives in surveillance videos. Current surveillance and control systems still require human monitoring and intervention. We present a system of automatic detection of weapons on video appropriate for surveillance and control purposes.

This work is an attempt to design and develop a system which can detect the pistols, revolver, other shot handheld weapons and objects that can most likely be confused with pistol class objects like wallets, cell phone, metal detector, selfie stick in no time with less computational resources. This work will indeed help in improving the security, law and order situation for the betterment and safety of humanity, especially for the countries who had suffered a lot with these kind of violent activities. Thus, our proposed system can also be implemented in CCTV to detect weapons or unsafe assets.

1.3 Significance of the Work

The main aim of this project is to detect the pistols, revolver, other shot handheld weapons and objects that can most likely be confused with pistol class objects like wallets, cell phone, metal detector, selfie stick in no time with less computational resources.

The main objective of this work is to develop a region proposal approach such that the image can be taken as the bounding boxes of input and output proposals related to all areas in a picture most probable to be the object. As this method takes a picture as the bounding boxes of input and output related to all patches in a picture most probable to be a category, so it proposes a region with the maximum score as the location of an object.

1.4 Objectives

The aim of proposed system is to detect the pistols, revolvers, other shot handheld weapons and objects that can most likely be confused with pistol class objects like wallets, cell phone, metal detector, selfie stick using Inception ResNet V2 and YOLOV4. The Objectives to be achieved are:

1. Detect a segment containing pistol, revolvers or other short handheld weapons with their probability of true in real-time.
2. Detect a segment containing non-pistol class objects with their probability of true in real-time.
3. Extraction of features for each segment.
4. Design a neural network based classifier that can detect both pistol and non-pistol classes.
5. Increase safety and anomaly event monitoring in crowded events or public places.

2. LITERATURE SURVEY

2.1. Review of the project

Z. Xiao, [12] proposed for automatic detection of the concealed pistols detected by passive millimeter in security applications. They extended four half-surrounded like features and use integral image to rapidly calculate the rectangle features. Then they obtained a multi-layer classifier cascaded by several strong classifiers using AdaBoost algorithm to detect the contraband. Various passive millimeter images from both published literatures and our own measurements are used for training and testing. The experimental results show that the metallic pistols in different sizes, shapes, and angles can be accurately detected, so this method is useful for automatic detection of pistols.

Sheen et al, [13] proposed CWD method based on a three-dimensional millimeter (mm) wave imaging method, for detecting hidden weapons at airports and other safe locations in the body. These techniques were derived from microwave holography techniques that utilize phase and amplitude information recorded over a two-dimensional aperture to reconstruct a focused image of the target. Millimeter-wave imaging is well suited for the detection of concealed weapons or other contraband carried on personnel since millimeter-waves are nonionizing, readily penetrate common clothing material, and are reflected from the human body and any concealed items. In this project, a wide-bandwidth three-dimensional holographic microwave imaging technique is described.

Z. Xue et al, [14] proposed a CWD technique based on a fusion-based technique of multi-scale decomposition, which combines color visual picture with infrared (IR) picture integration. Here a large set of existing image fusion algorithms are identified and their performance is compared for the CWD application using quantitative and qualitative measures.

R. Blum et al, [15] proposed a CWD method based on the inclusion of visual picture and IR or mm wave picture using a multi-resolution mosaic technique to highlight the hidden weapon of the target picture. Here Images acquired by heterogeneous image sensors may provide complementary information about the scene, and a so-called multiresolution image fusion technique is applied to integrate the information at the pixel level. In this study, the MR mosaic

technique is employed to combine the visual and IR or mm wave images, where the concealed weapon is first detected by a fuzzy k-means clustering method from the IR or mm wave image.

E. M. Upadhyay et. Al, [16] proposed a CWD technique using image fusion. They used IR image and visual fusion to detect hidden weapons in a situation where the image of the scene was present over and under exposed area. He proposed a novel algorithm to generate one multi exposed - multi modal image from a set of visual images with multiple exposures and a set of infrared images with multiple exposures. Their methodology was to apply a homomorphic filter captured at distinct exposure conditions to visual and IR pictures. The method employs homomorphic filter, entropy of blocks and blending approach for image fusion. Experiments were performed on different sets of multi exposed images by varying the block size and the blending parameter which shows that the proposed method outperforms the results obtained without homomorphic filter.

R. Chellapa et.al, [22] discussed briefly object tracking and detection in surveillance cameras. The authors had explained the tracking of an object using multiple surveillance cameras. They first discussed the nature of the challenges in the context of visual sensor networks. Then, they showed how real-world constraints can be favorably exploited in order to tackle the challenges. Examples of real-world constraints are (a) the presence of a world plane, (b) the presence of a three-dimiensional scene model, (c) consistency of motion across cameras, and (d) color and texture properties. In this regard, the main focus of this project is towards highlighting the efficient use of the geometric constraints induced by the imaging devices to derive distributed algorithms for target detection, tracking, and recognition. Their discussions are supported by several examples drawn from real applications. Lastly, they also describe several potential research problems that remain to be addressed.

S. Zhang, [23] proposed a framework for achieving these tasks in a nonoverlapping multiple camera network. A new object detection algorithm using mean shift (MS) segmentation is introduced, and occluded objects are further separated with the help of depth information derived from stereo vision. The detected objects are then tracked by a new object tracking algorithm using a novel Bayesian Kalman filter with simplified Gaussian mixture (BKF-SGM). It employs a Gaussian mixture (GM) representation of the state and noise densities and a novel

direct density simplifying algorithm for avoiding the exponential complexity growth of conventional Kalman filters (KFs) using GM.

J.S Marques, [24] proposed distinct techniques for evaluating the efficiency of distinct algorithms for object recognition. They proposed novel methods to evaluate the performance of object detection algorithms in video sequences. This procedure allows us to highlight characteristics (e.g., region splitting or merging) which are specific of the method being used. The proposed framework compares the output of the algorithm with the ground truth and measures the differences according to objective metrics. In this way it is possible to perform a fair comparison among different methods, evaluating their strengths and weaknesses and allowing the user to perform a reliable choice of the best method for a specific application. We apply this methodology to segmentation algorithms recently proposed and describe their performance. These methods were evaluated in order to assess how well they can detect moving regions in an outdoor scene in fixed-camera situations.

B. Triggs et.al, [25] described histogram oriented gradient (HOG). HOG became a novel architecture for feature extraction. It was used mostly in applications involved in human detection. They studied the influence of each stage of the computation on performance, concluding that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping descriptor blocks are all important for good results. The new approach gives near-perfect separation on the original MIT pedestrian database, so we introduce a more challenging dataset containing over 1800 annotated human images with a large range of pose variations and backgrounds.

C. Anagnostopoulos, [26] proposed a new algorithm for vehicle license plate identification is proposed, on the basis of a novel adaptive image segmentation technique (sliding windows) in conjunction with a character recognition neural network. The algorithm was tested with 2820 natural scene gray level vehicle images of different backgrounds and ambient illumination. The camera focused on the plate, while the angle of view and the distance from the vehicle varied according to the experimental setup. The license plates properly segmented were 2719 over 2820 input images (96.4%). The optical character recognition (OCR) system is a two layer probabilistic neural network with topology 108-180-36, whose performance reached 97.4%. The PNN was trained to identify multi-font alphanumeric characters from car license plates based on data obtained from algorithmic image processing.

M. Grega, [27] proposed an algorithm that is capable of detecting a person carrying an uncovered firearm and alerting the CCTV operator of a potentially dangerous situation. They presented the limitations and difficulties for such an image analysis application, and discussed the construction of the proposed algorithm, and show the numerical results in terms of sensitivity and specificity.

L. Ward et al, [28], [29] was the first to detect firearms in 2007 and a surveillance system was also implemented by them a year later in 2008. In the aforementioned work, writers created an accurate pistol detection model for RGB pictures. However, in the same scene, their method did not detect various pistols [11], [10], [29]. The approach used comprises of first removing non-related items from the segmented picture using the K-mean clustering algorithm and then applying the SURF (Speed up Robust Features) method to detect points of interest. Darker gave the concept of SIFT based weapon detection algorithm and for ROI estimation, used the motion segmentation method [30]. SIFT algorithm is prone to false alarms, so for estimating ROI, authors used motion segmentation rather than using SIFT on complete image. When ROI was determined, then SIFT was applied to detect firearms in their case.

X. Zhang et al, [33] proposed a residual learning framework to ease the training of networks that are substantially deeper than those used previously. They explicitly reformulated the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. They concluded an important finding that helped my work. They concluded that the automatic feature representation gave improved results rather than manual features. Not only the learned features were better in performance, they also had learned the deep representation of the data and reduced a lot of manual work, and saved time and energy.

Rohith Vajhala et al, [34] proposed the technique of knife and gun detection in surveillance systems. They had used HOG as a feature extractor along with backpropagation of artificial neural networks for classification purposes. In this work the detection of weapon from CCTV frame is acquired by using Histogram of Oriented Gradients (HOG) as feature vector artificial neural networks performing back-propagation algorithm for classification. Weapon detection has been performed using three methods. In the first method, the weapon in the image is detected directly with-out human detection. Second and third methods use HOG and back-ground subtraction methods for detection of human prior to detection of a weapon. A knife and a gun are considered as weapons of interest in this work. The performance of the proposed

detection methods was analyzed on test image dataset containing knives, guns and im-ages without weapon. The accuracy rate 846% has been achieved by a single-class classier for knife detection. A gun and a knife have been detected by the three-class classier with an accuracy rate 83.0%.

M. Nakib, [35] proposed CNN (Convolutional Neural Network) in the use of detect knife, blood and gun from an image. Detecting these threatening objects from image can give us a prediction whether a crime occurred or not and from where the image is taken. They emphasized on the accuracy of detection so that it hardly gives us wrong alert to ensure efficient use of the system. This model use Rectified Linear Unit (ReLU), Convolutional Layer, Fully connected layer and dropout function of CNN to reach a result for the detection. They used Tensorflow, a open source platform to implement CNN to achieve our expected output. The proposed model achieves 90.2% accuracy for the tested dataset.

Verma et al, [36] had also used the deep learning technique to detect weapons and used the Faster RCNN model. The work was performed on imfdb, which in my opinion is not suitable to train a model for real-time case. They claimed to have an accuracy of 93.1% on that dataset but in the case of weapon detection, only achieving higher accuracy is not enough, and precision and recall must the considered.

Siham Tabik et al, [37] work was very much related to the real-time scenario. They used Faster RCNN to detect weapons in real-time using sliding window and region proposal methods. This work presents a novel automatic handgun detection system in videos appropriate for both, surveillance and control purposes. They reformulated this detection problem into the problem of minimizing false positives and solve it by building the key training data-set guided by the results of a deep Convolutional Neural Networks (CNN) classifier, then assessing the best classification model under two approaches, the sliding window approach and region proposal approach. Best results were obtained by using the region proposal technique. The sliding window was also very time consuming and took 14 s/image, on the other hand, the region proposal method processed the image in 140ms with 7 fps.

Javed Iqbal et al, [38] proposed orientation aware detection of the object. This system is more suitable for long and thin objects like rifles etc. Images of very high quality were used for training and testing purposes, which may make it less suitable for real-time scenarios. They

proposed a weakly supervised Orientation Aware Object Detection (OAOD) algorithm which learns to detect oriented object bounding boxes (OBB) while using AxisAligned Bounding Boxes (AABB) for training. The proposed OAOD is different from the existing oriented object detectors which strictly require OBB during training which may not always be present. The proposed ITU Firearm dataset (ITUF) contains a wide range of guns and rifles. The OAOD algorithm is evaluated on the ITUF dataset and compared with current state-of-the-art object detectors, including fully supervised oriented object detectors. OAOD has outperformed both types of object detectors with a significant margin. The experimental results (mAP: 88.3 on AABB & mAP: 77.5 on OBB) demonstrate the effectiveness of the proposed algorithm for firearm detection.

Jose Luis Salazar Gonz'alez et al, [39] work was very much related to achieve real-time results. This article presents a new dataset obtained from a real CCTV installed in a university and the generation of synthetic images, to which Faster R-CNN was applied using Feature Pyramid Network with ResNet-50 resulting in a weapon detection model able to be used in quasi real-time CCTV (90 ms of inference time with an NVIDIA GeForce GTX-1080Ti card) improving the state of the art on weapon detection in a two stages training. In this work, an exhaustive experimental study of the detector with these datasets was performed, showing the impact of synthetic datasets on the training of weapons detection systems, as well as the main limitations that these systems present nowadays. The generated synthetic dataset and the real CCTV dataset are available to the whole research community. They did immense experimentation using different datasets and trained Faster–RCNN using Feature Pyramid Network with Resnet50 and improves the previous state of the art by 3.91 %.

K. Simonyan and A. Zisserman, [41] worked on the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Their main contribution is a thorough evaluation of networks of increasing depth using an architecture with very small (3x3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16-19 weight layers. These findings were the basis of our ImageNet Challenge 2014 submission, where their team secured the first and the second places in the localisation and classification tracks respectively. They also showed that their representations generalise well to other datasets, where they achieve state-of-the-art results.

They have made their two best-performing ConvNet models publicly available to facilitate further research on the use of deep visual representations in computer vision.

2.2. Limitations of the existing systems

1. These algorithms were slow for real-time scenarios with 14s per image.
2. Although these classifiers gave good accuracies, the slowness of the sliding window method was a big problem, especially for the realtime implementation purpose

3. SYSTEM ANALYSIS

3.1. Requirement Specification

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed.

In software engineering, such requirements are often called functional specifications. Requirements analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

3.1.1 FUNCTIONAL REQUIREMENTS

Functional requirements define the internal workings of the software: that is, the technical details, data manipulation and processing and other specific functionality that show how the use cases are to be satisfied. They are supported by non-functional requirements, which impose constraints on the design or implementation.

3.1.1.1 Hardware Requirements

- SYSTEM : Intel Core i5-2600
- RAM : 4 GB or more

3.1.1.2 Software Requirements

- O/S : Windows 7/8/10
- Platform: Python idle 3.7 version
- Language : Python

3.1.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are requirements which specify criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that specify specific behavior or functions. Typical non-functional requirements are reliability, scalability, and cost. Non-functional requirements are often called

the ilities of a system. Other terms for non-functional requirements are "constraints", "quality attributes" and "quality of service requirements".

Reliability: If any exceptions occur during the execution of the software it should be caught and thereby prevent the system from crashing.

Scalability: The system should be developed in such a way that new modules and functionalities can be added, thereby facilitating system evolution.

Cost: The cost should be low because a free availability of software package.

3.2. UML Diagrams for the project

UML is an acronym that stands for Unified Modeling Language. Simply put UML is a modern approach to modeling and documenting software. In fact, it's one of the most popular business process modeling techniques.

It is based on diagrammatic representations of software components. As the old proverb says: "a picture is worth a thousand words". By using visual representations, we are able to better understand possible flaws or errors in software or business processes.

The elements are like components which can be associated in different ways to make a complete UML picture, which is known as diagram. Thus, it is very impor- tant to understand the different diagrams to implement the knowledge in real-life systems.

Any complex system is best understood by making some kind of diagrams or pictures. These diagrams have a better impact on our understanding. If we look around, we will realize that the diagrams are not a new concept but it is used widely in different forms in different industries.

The various UML diagrams are

1. Usecase diagram
2. Activity diagram
3. Sequence diagram
4. Colloboration diagram
5. Object diagram
6. State chart diagram
7. Class diagram

8.Component diagram

9.Deployment diagram

3.2.1 Use Case View

3.2.1.1. Identification of Actors

A use case diagram is a graph of actors, a set of use cases enclosed by a system boundary, communication (participation) associations between the actors and users and generalization among use cases. The use case model defines the outside (actors) and inside (use case) of the system's behavior. Actors are not part of the system. Actors represent anyone or anything that interacts with (input to orreceive output from) the system. Use-case diagrams can be used during analysis to capture the system requirements and to understand how the system should work. During the design phase, you can use use-case diagrams to specify the behavior of the system as implemented. Use case is a sequence of transactions performed by a system that yields a measurable result of values for a particular actor. The use cases are all the ways the system may be used.

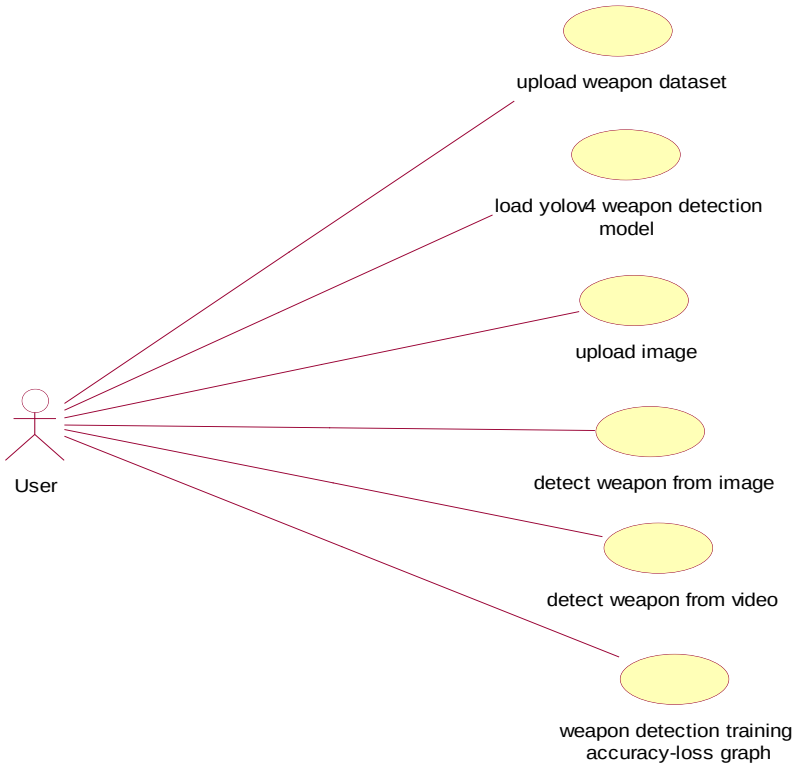


Fig3.1: Use-Case Diagram

3.2.1.2. Activity Diagram

An Activity diagram is a variation of a special case of a state machine, in which the states are activities representing the performance of operations and the transitions are triggered by the completion of the operations. The purpose of Activity diagram is to provide a view of flows and what is going on inside a use case or among several classes. Activity diagrams contain activities, transitions between the activities, decision points, and synchronization bars. An activity represents the performance of some behavior in the workflow. In the UML, activities are represented as rectangles with rounded edges, transitions are drawn as directed arrows, decision points are shown as diamonds, and synchronization bars are drawn as thick horizontal or vertical bars as shown in the following. The activity icon appears as a rectangle with rounded ends with a name and a component for actions.

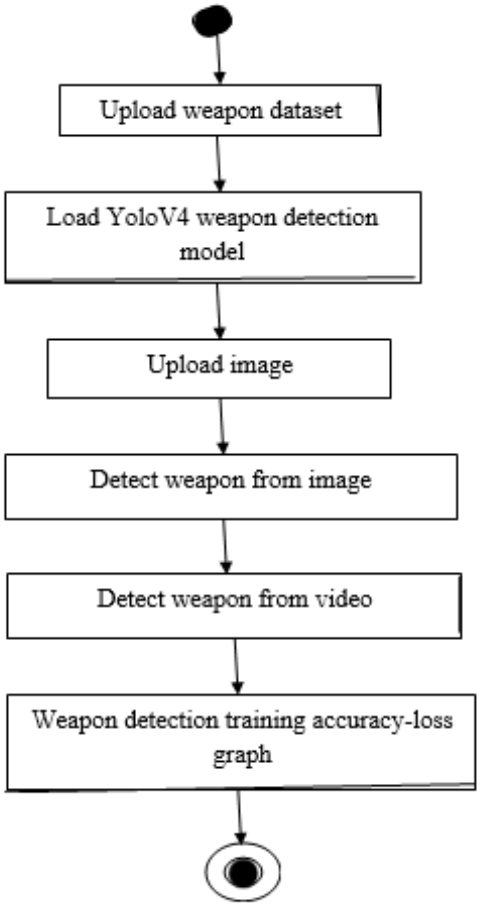


Fig3.2: Activity Diagram for the System

3.2.1.3. Sequence Diagram

A sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called as event diagrams.

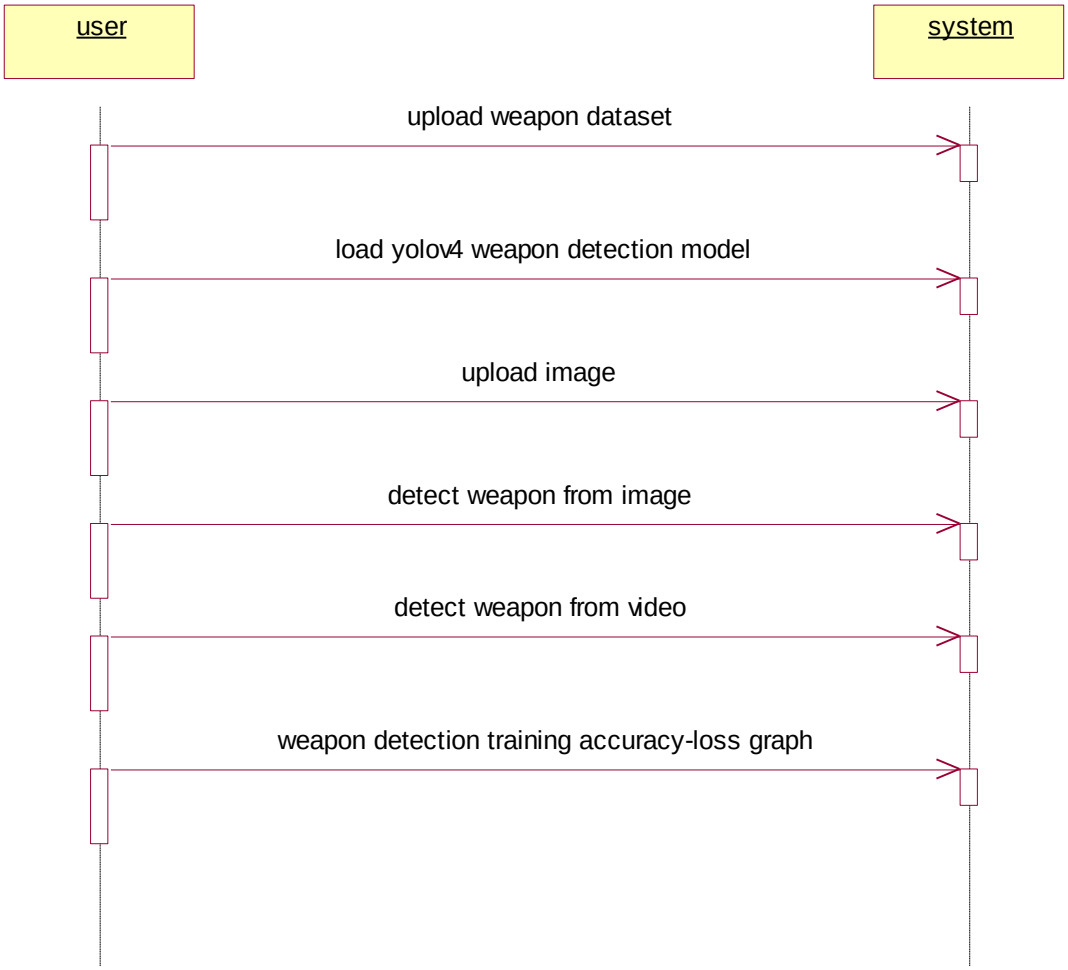


Fig3.3: Sequence Diagram for the System

3.2.1.4. Collaboration Diagram

A collaboration diagram shows that the order of messages that implement an operation or a transaction. Collaboration diagrams show objects, their links, and their messages. They can also contain simple class instances and class utility instances. Each collaboration diagram provides a view of the interactions or structural relationships that occur between objects and object like entities in the current model. Collaboration diagrams and sequence diagrams are called interaction diagrams. A collaboration diagram shows that the order of messages that implement an operation or a transaction. Collaboration diagrams show objects, their links, and their messages. They can also contain simple class instances and class utility instances. Each collaboration diagram provides a view of the interactions or structural relationships that occur between objects and object like entities in the current model.

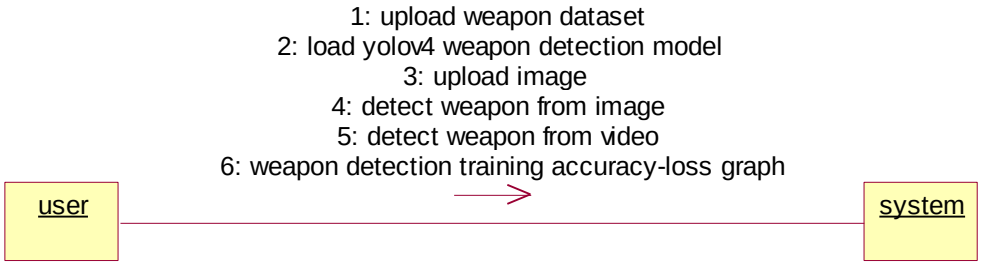


Fig3.4: Collaboration Diagram

3.2.1.5. Class Diagram

A Class is a description of a group of objects with common properties (attributes)common behavior(operations),common relationships too their objects, and common semantics. Thus, a class is a template to create objects. Each object is an instance of some class and objects cannot be instances of more than one class. In the UML, classes are represented as compartmentalized rectangles.

Class diagrams contain icons representing classes, interfaces, and their relationships. You can create one or more class diagrams to represent the classes at the top level of the current model; such class diagrams are themselves contained by the top level of the current model. You can also create one or more class diagrams to represent classes contained by each package in your model such class diagrams are themselves contained by the package enclosing the classes they represent the icons representing logical packages and classes in class diagrams.

- The top compartment contains the name of the class.
- The middle compartment contains the structure of the class(attributes).
- The bottom compartment contains the behavior of the class(operations).

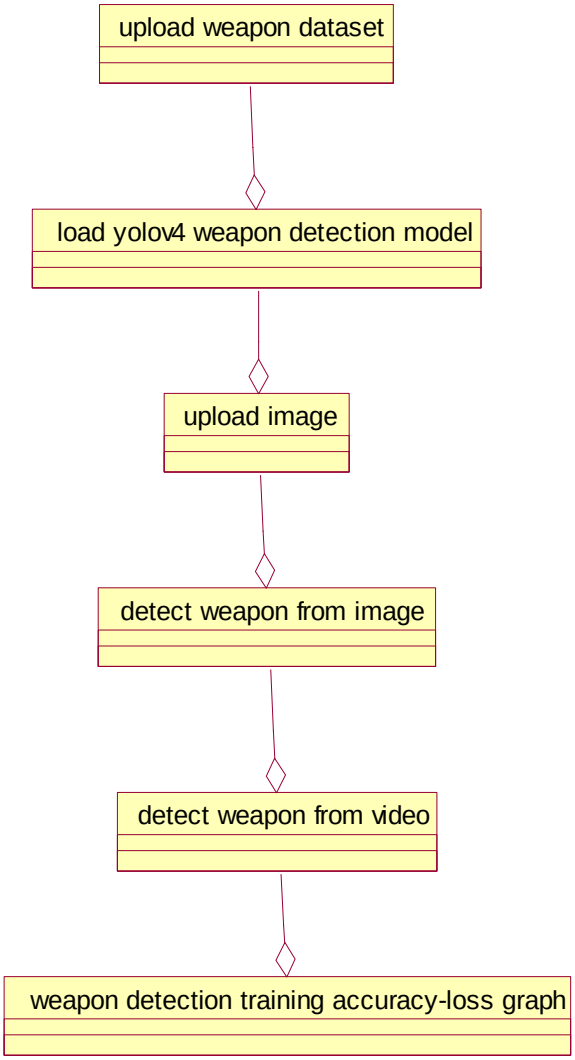


Fig3.5: Class Diagram for the System

3.2.1.6. State Chart Diagram

State chart diagram cannot be created for every class in the system, it is only for those lass objects with significant behavior. State transition:

A state transition indicates that an object in the source state will perform certain specified actions and enter the destination state when a specified event occurs or when certain conditions are satisfied. A state transition is a relationship between two states, two activities, or between an activity and a state.

We can show one or more state transitions from a state as long as each transition is unique. Transitions originating from a state cannot have the same event, unless there are conditions on the event.

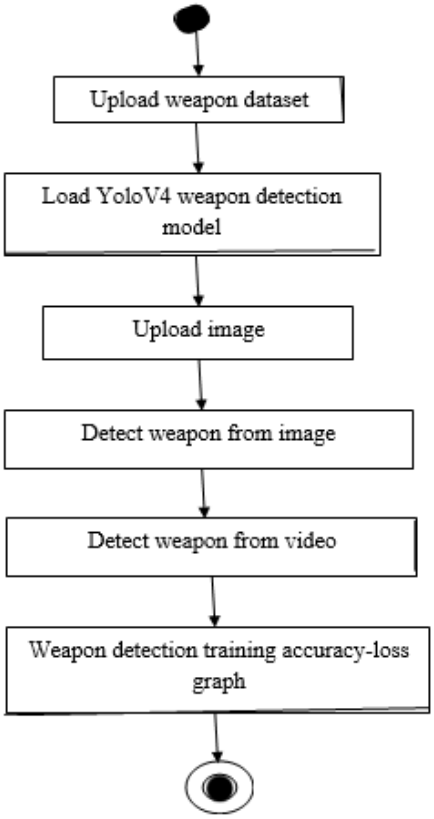


Fig3.6: State Chart Diagram

3.2.1.7. Component Diagram

Component Diagrams show the dependencies between software components in the system. The nature of these dependencies will depend on the language or languages used for the development and may exist at compile-time or atruntime.

A component diagram shows the allocation of classes and objects to components in the physical design of a system. A component diagram may represent all or part of the component

architecture of a system along with dependency relationships. The dependency relationship indicates that one entity in a component diagram uses the services or facilities of another.

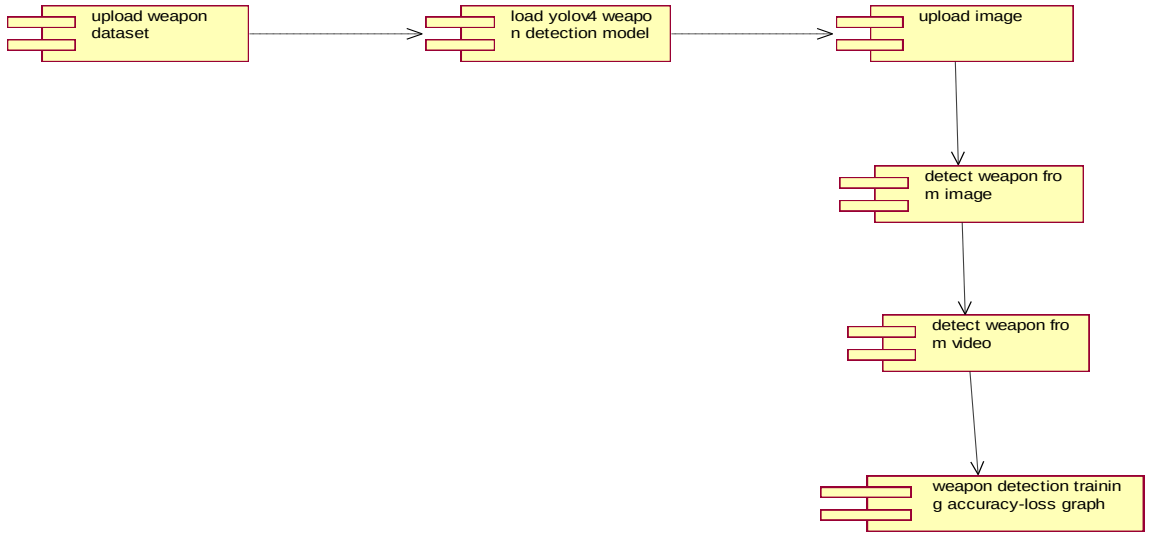


Fig 3.7: Component Diagram

3.2.1.8. Deployment Diagram

The second type of implementation diagram provided by UML is the deployment diagram. Deployment diagrams are used to show the configuration of run-time processing elements and the software components and processes that are located on them. Deployment diagrams are made up of nodes and communication associations.

Nodes are typically used to show computers and the communication associations show the network and protocols that are used to communicate between nodes. Nodes can be used to show other processing resources such as people or mechanical resources. Nodes are drawn as 3D views of cubes or rectangular prisms, and the following figure shows a simplest deployment diagram where the nodes connected by communication associations.

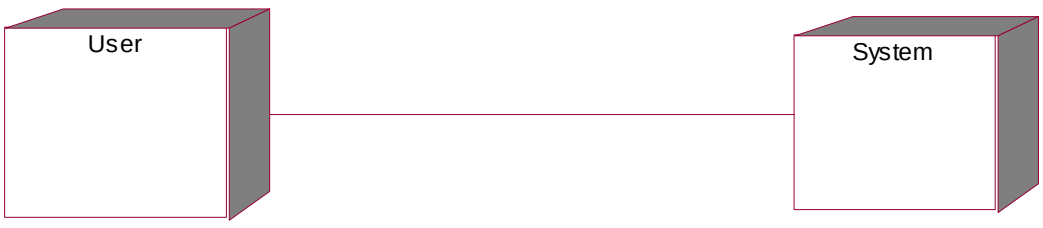


Fig3.8: Deployment Diagram

4. SYSTEM DESIGN

4.1. Architecture of the System

The methodology adopted in this work features the state of art deep learning, especially the convolutional neural networks due to their exceptional performance in this field. [40]. The aforementioned techniques are used for both the classification as well as localizing the specific object in a frame so both the object classification and detection algorithms were used and because our object is small with other object in background so after experimentation we found the best algorithm for our case. Sliding window/classification and region proposal/object detection algorithms were used, and these techniques will be discussed later in this section.

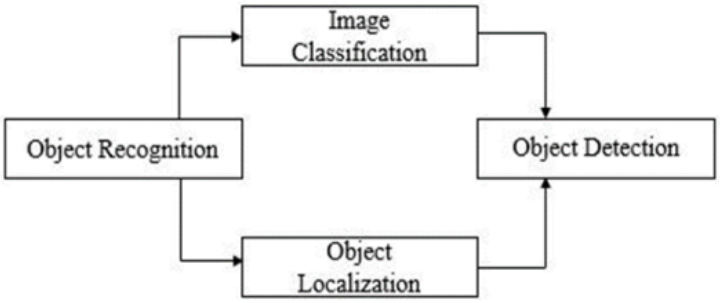


Figure 4.1: Object Recognition to detection Hierarchy

The general methodology used in training and optimization. It starts with defining a problem, finding the required dataset, applying pre-processing methods, and then finally training and evaluating the dataset. If the evaluation is correct then we save those weights as a classifier but if it's incorrect then comes the process of backpropagation algorithm along with the gradient descent algorithm [60]. In backpropagation, weights are optimized by subtracting the partial derivative of cost function J(O) with a multiplier of the learning rate alpha α from the old or previous weight value. Gradient descent is the main weight optimization algorithm. It is used as a base in all optimizers used for the modeling and it helps in converging the model and reaching the minima where we get the best and desired weights values.

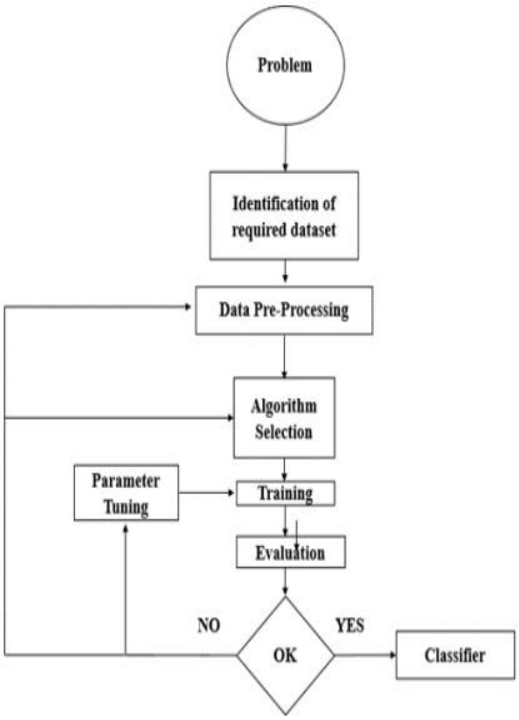


Figure 4.2: Training and Optimization Flow Diagram

4.2. Proposed System

REGION PROPOSAL/OBJECT DETECTION MODELS:

This technique takes an image as the bounding boxes of input and output proposals related to all areas in a picture most probable to be the object. These regional proposals may be noisy; coinciding not containing the object flawlessly, but there is a proposal among these region proposals related to the original target object. As this method takes a picture as the bounding boxes of input and output related to all patches in a picture most probable to be a category, so it proposes a region with the maximum score as the location of an object. Instead of considering all possible regions of the input frame as possibilities, this method uses detection proposal techniques to select regions [57]. Region-based CNNs (R-CNN) was the first detection model to introduce CNNs under this approach [58]. The selective search method of this approach produces 2000 boxes having maximum likelihood.

The object detectors used for real-time detection are:

- SSD MobilNetV1

- YoloV3
- Faster RCNN-Inception ResNetV2
- YoloV4

YOLOV4:

YOLOv4 is known for its up-gradation in terms of AP and FPS. YOLOv4 prioritizes real-time object detection and training takes place on a single CPU. YOLOv4 has obtained state-of-art results on the COCO dataset with 43.5% speed (AP) at 65 Performance (FPS) on Tesla V100. This achievement is the result of a combination of the features like DropBlock Regularization, Data Augmentation, Mish-Activation, CrossStage-Partial-connections (CSP), Self-adversarial-training (SAT), Weighted-Residual-Connections (WRC) and many more. There are two types of models, one and two-staged object detectors. In two-stage detectors works in two parts that are first regions of importance are detected and then regions are classified to see if the object is detected in that particular region. YOLOv4 being a single staged object detector works more accurate and faster than Two staged detectors like R-CNN, Fast R-CNN.

YOLOv4 basically uses one of the three models as its backbone. Three feature extractor models include:

CSPResNext50

Both CSPResNext50 and CSPDarknet53 are DenseNet based models. It works similar to CSPDarknet53 that operates on the CSPNet strategy. Considering the COCO dataset, CSPDarknet53 is better in classifying objects than CSPResNext50. CSPResNext50 consists of 16 CNN layers with 425×425 receptive field and 20.6 M parameters while CSPDarknet53 consists of 29 CNN layers with 725×725 receptive field and 27.6 M parameters.

CSPDarknet53

It is a widely used Backbone of Object detection that makes use of DarkNet-53. YOLOv4 Specifically uses CSPDarknet53 as its backbone. It operates on a CSPNet strategy of dividing DenseBlock consisting feature map in two halves and then merging them together via cross-stage hierarchy. The former part circumvents the base layer and is used as input of the next transition layer. The later part of the base layer undergoes DenseBlock. This strategy decreases the computational complexity. It has an accuracy that is better when compared to other ResNet models and hence has good performance.

EfficientNet-B3

It is used as an image Classification Model that is particularly used to attain state-of-art accuracy. It is generally utilized to restudy the Convolutional Neural Network scaling and is based on AutoML. AutoML mobile Framework was developed in order to develop a small-sized network known as EfficientNet-B0. The Compounding Scaling as the name suggests helps in scaling up AutoML baseline in order to gain Efficient-B1 to Efficient-B7

4.3 DataSet

To train all those algorithms there is no public dataset available so author of this project is creating his own dataset from CCTV images. We also downloaded few KNIVES and GUNS images from Google to train algorithms.

Training all algorithms may take days of time and it's not possible to train all algorithms so we have trained YoloV4 model and then this model can be applied on images and videos to detect weapon.

Below screen showing images from dataset used to train YOLO algorithm

DATASET

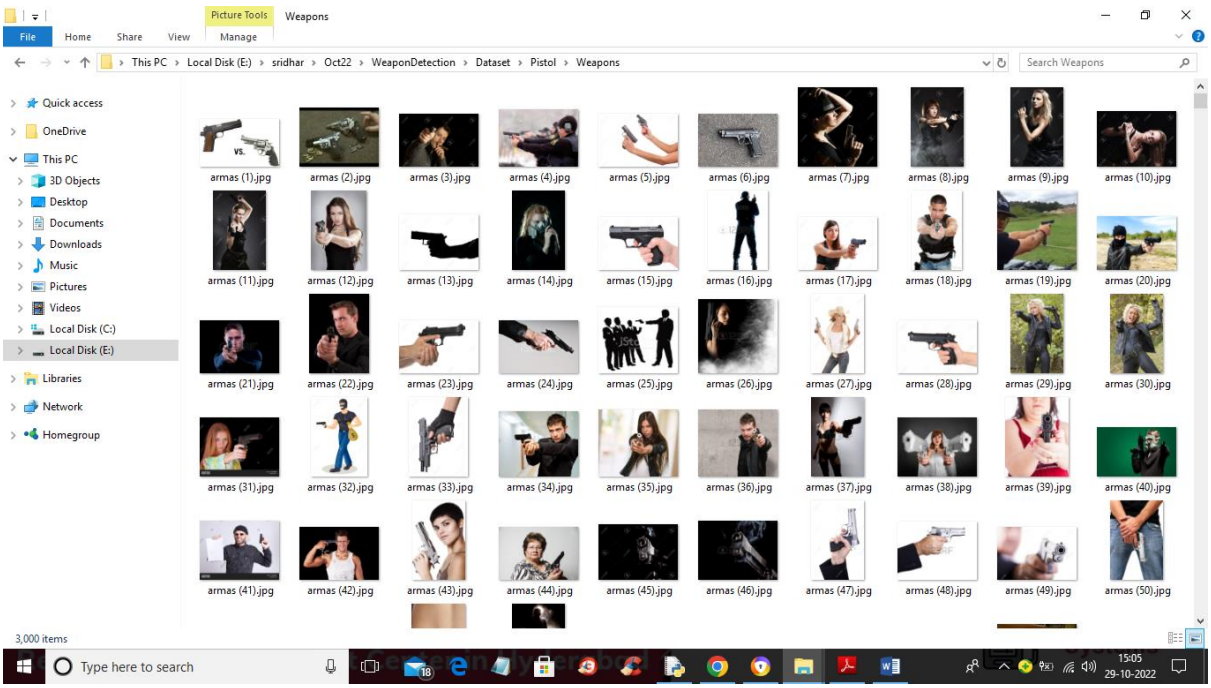


Figure 4.3: Dataset

5. IMPLEMENTATION

5.1. Algorithms

YOLOV4:

YOLOv4 is known for its up-gradation in terms of AP and FPS. YOLOv4 prioritizes real-time object detection and training takes place on a single CPU. YOLOv4 has obtained state-of-art results on the COCO dataset with 43.5% speed (AP) at 65 Performance (FPS) on Tesla V100. This achievement is the result of a combination of the features like DropBlock Regularization, Data Augmentation, Mish-Activation, CrossStage-Partial-connections (CSP), Self-adversarial-training (SAT), Weighted-Residual-Connections (WRC) and many more. There are two types of models, one and two-staged object detectors. In two-stage detectors works in two parts that are first regions of importance are detected and then regions are classified to see if the object is detected in that particular region. YOLOv4 being a single staged object detector works more accurate and faster than Two staged detectors like R-CNN, Fast R-CNN.

DISCUSSION ON PROPOSED RESULTS

5.2 . Experimental Setup

To implement this project we have designed following modules

1) Upload Weapon Dataset: using this module we will upload dataset to application
2) Load Yolov4 Weapon Detection Model: this will read dataset images and then trained or load Yolov4 model and then calculate its prediction accuracy
3) Upload Image: using this module we will upload test image
4) Detect Weapon from Image: using this module we will apply Yolov4 model on loaded image to detect weapon
5) Detect Weapon from Video: using this module we will upload video and then YoloV4 will analyse each frame in the video to detect weapon
6) Weapon Detection Training Accuracy-Loss Graph: using this module we will plot YoloV4 training accuracy and loss graph

5.3. Predicted Results

5.3.1 DATA PRE-PROCESSING AND ANNOTATION:

Many things affect the performance of a Machine Learning (ML) model for a specified job. First, the representation and quality of the data are essential. If there are many irrelevant and redundant data existing or noisy and unreliable data, then it is harder to discover representation during the training stage. Data preparation and filtering steps take significant processing time in ML issues [61]. The pre-processing process involves data cleaning, standardization, processing, extraction and choice of features, etc. The final training dataset is the result of pre-processing processes applied to the collected dataset. Pre-processing is necessary for better training of a model, so the first step is to make the same size or resolution of the dataset. The next step is to apply the mean normalization. The third step is making bounding boxes on these images, which is also called annotation, localization, or labeling. In data, labeling a bounding box is made on each image. The value x, y coordinates, and width, height of the labeled object was stored in xml, csv or txt format. Following are the four main steps of data preprocessing:

- Image scaling
- Data-augmentation
- Image labeling
- Image Filtering using OpenCV
  - RGB to Grayscale
  - Equalized
  - Clahe

5.4.2 EXPERIMENTATION AND RESULTS:

We have detected weapons in real-time CCTV streams in low resolution, dark light with real-time frame per second. Most of the work done before was on detecting images and videos of high quality and because those models were trained on high-quality datasets, it is not possible to then detect an object of low resolution in real-time. The results are analyzed after training and testing models on dataset. As described in the methodology section the results for different approaches are evaluated. Our main problem statement is of real-time detection because 97% of

weapon used in robbery cases were pistol or revolver, so different dataset results have been evaluated here for sliding window and region proposal approach. The performance of these models was analyzed by comparing them in terms of the standard metrics of F1-score and frame per seconds along with mean average precision (mAP) for the best performed model and these terms are calculated by using the below equation 1,2 and 3. F1 score is ratio of the precision and recall functions.

$$\text{Precision} = \text{True Positives} / (\text{True Positives} + \text{False Positives}) \quad (1)$$

$$\text{Recall} = \text{True Positives} / (\text{True Positives} + \text{False Negatives}) \quad (2)$$

$$\text{F1-score} = 2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall}) \quad (3)$$

5.4 Comparison of predicted model with other model

After experimentation on the previous two datasets and not finding satisfactory results for the real-time case a new dataset was made. Images were collected from robbery videos, our own dataset images holding a weapon in different scenarios, images with a dark background and low resolution, and images extracted from applying different OpenCV filters are added to make real-time detection possible. A total of 3327 images are used in this case. Following object detection models were trained and evaluated using this dataset:

- SSD MobilNetV1
- YoloV3
- Faster RCNN-Inception ResNetV2
- YoloV4

Table 5.1 below shows the results for the aforementioned detection models for this dataset at a standard threshold score of 50%. Yolov4 performs best among all the models of both the sliding window and region proposal approach. Performance graph for yolov4 in terms of loss and mean average precision (mAP) on a validation dataset is shown in Fig. 5.1.

Table 5.1: Region Proposal/Object Detection Models

Sr. No	Models	IoU Threshold=50%		
		Precision	Recall	F1-score
1	SSD-MobileNet-v1	62.79%	60.23%	59%
2	Yolov3	85.86%	87.34%	86%
3	FasterRcnn-InceptionResNetV2	86.38%	89.25%	87%
4	Yolov4	93%	88%	91%

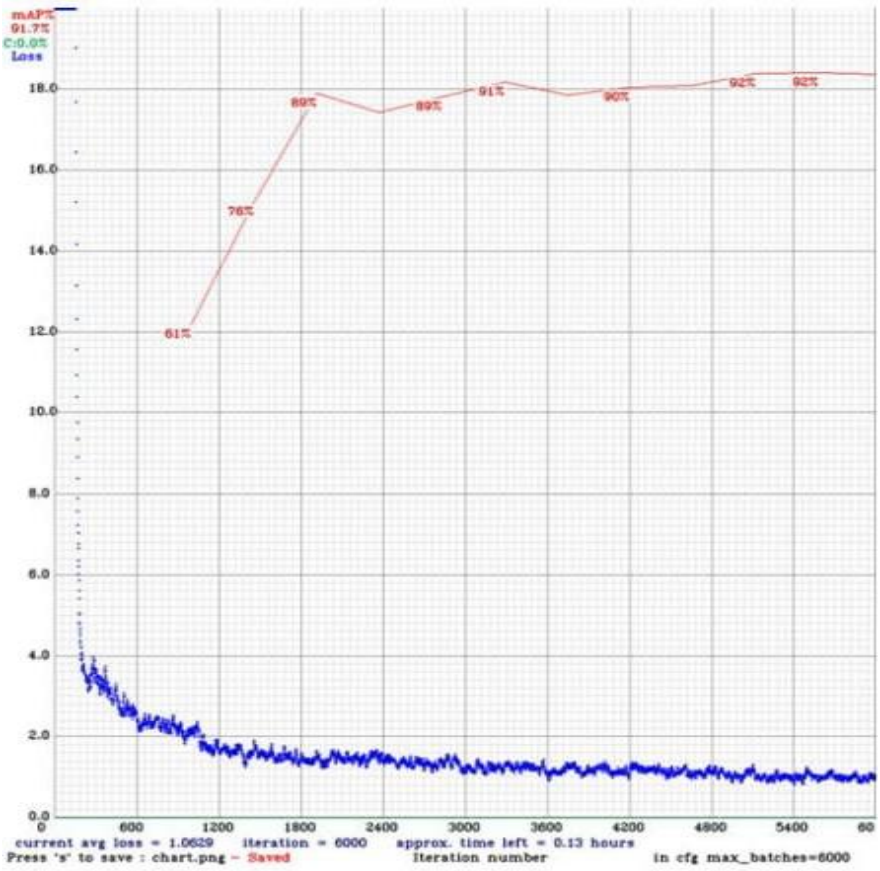


Figure 5.1: Best Object Detection Model-Yolov4: loss vs Map

We can see that how smooth is the model loss curve and how precisely it converges to the best level giving a very good loss score of 1.062 and a mean average precision of 91.73%. The mean average precision is the mean of the average precision values for all the relevant classes. The associated values of average precision (AP) for pistol and not-pistol class for the calculation of mean average precision value is given in Table 5.2.

Table 5.2: Best Performed Model Yolov4: mAP Calculation

IoU_ Threshold	Average Precision Pistol (AP_P)	Average Precision Pistol (AP_{NP})	Mean Average Precision (mAP)
0.35	96.91%	92.41%	94.66%
0.50	94.00%	89.47%	91.73%
0.75	60.17%	67.41%	63.79%

The mean average precision value is calculated for the yolov4 model as it performs best in all scenario and accurately detected the desired object even when the object has a very small presense in the frame and there were lots of other objects in the background as well.

4.4. ANALYSIS AND DISCUSSION

- Table 5.1, 5.2, and 5.3 above shows the comparison between the classification and object detection models usingstandard metrics of precision, recall, and F1-score for evaluation.
- Some classification models showed good results but they were not suitable for a real-time scenario, were slow, not much accurate, and fast as compared to the object detection models as they performs very well and achieved high precision and recall.
- The reason why some classification models have a good F1-score is the training and evaluation on initial datasets we made when starting this work, but after experimentation, we come to know that these models are not suitable for real-time scenarios having the background objects. Object detection models performed well for the real-time scenario and performance comparison in terms of speed and F1-score between the detection models can

be seen from Fig. 5.3. Inference results are obtained using the NVIDIA RTX 2080ti for each model.

- The standard metrics of mean average precision (mAP), recall and F1-score are calculate and all the models have been compared at a benchmark IoU threshold of 0.50 or 50%.
- Yolov4 performs best amongst all models with a mean average precision and F1-score of 91.73% and 91% respectively with detection confidence of 99% in the majority of cases.
- Comparison in terms of test accuracy vs F1-score for the best-performed models of both classification and detection approaches is shown in the Fig.4.6. Accuracy and F1-score for VGG, Inceptionv3, InceptionResNetv2, SSDMobileNet, FasterRCNN-InceptionResNetv2, Yolov3 and yolov4 are 78.20%, 85.20%, 92.20%, 79%, 96%, 94%, 99% and 81.69%, 84.36%, 85.74%, 59%, 87%, 86% and 91% respectively.

It is very hard to do a comparison with studies conducted previously on this subject because each study has its own dataset, models and metrics used to evaluate performance. It should also be noticed achieve realtime detection, we also need to have a realtime dataset for traning because with high quality training images we cannot achieve results in realtime. Each study also has different testing conditions, either just on images, videos or on images with high quality but our approach from start was to achieve realtime results. In some studies, the performanece metric used is accuray, others have precisoin or mean average precision (mAP) but mostly mAP is used as standard so we have given comparison resultsin terms of mAP and precisoin at a standard iou threshold of 50%, which ever was available.

Table 5.3: Comparison with some existing studies

Study	Algorithm	Precision	mAP
Javed Iqbal et al. 2019	OOAD	N/A	85.40%
Roberto Olmos et al. 2017	Faster RCNN	84.20%	N/A
Jose Luis Salazar Gonz´alez et al. 2020	Faster RCNN using FPN	88.23%	N/A
Ours	Yolov4	93 %	91.73%

6. TESTING

6.1 Objective of Testing

Testing is a fault detection technique that tries to create failure and erroneous states in a planned way. This allows the developer to detect failures in the system before it is released to the customer.

Note that this definition of testing implies that a successful test is test that identifies faults. We will use this definition throughout the definition phase. Another often used definition of testing is that it demonstrates that faults are not present.

Testing can be done in two ways:

- Top down approach.
- Bottom up approach

Top-down approach –

This type of testing starts from upper level modules. Since the detailed activities usually performed in the lower level routines are not provided stubs are written.

Bottom-up approach –

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded within the larger system. In this project, bottom-up approach is used where the lower level modules are tested first and the next ones having much data in them.

6.2 Testing Methodologies

- Unit testing
- Integration testing
- User Acceptance testing
- Output testing
- Validation testing

Unit testing: Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring

that it functions properly as a unit. Hence, the naming is Unit Testing. During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

Integration testing: Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

User Acceptance testing: User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

Output testing: After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways - one is on screen and another in printed format.

Validation testing: Validation checks are performed on the following fields:

Text field: The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always ashes and error message.

Numeric field: The numeric field can contain only numbers from 0 to 9. An entry of any character ashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested. A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

Preparation of test data: Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

Using Live test data: Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves. It is difficult to obtain live data in sufficient amounts to conductive extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

Using artificial test data: Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program. The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an in dependent team of testers formulates a testing plan, using the systems specifications.

6.3 Test Cases

Table 6.1 :Test Case for Upload Weapon Dataset

Test Case Id	Test Scenario	Test Case	Pre Condition	Test Steps	Test Data	Expected	Post Condition	Actual Result	Test Status (P/F)
1	Upload Weapon Dataset	Preprocessing	Avail-ability of Dataset	Making the images available for model	dataset	No Error	Image prepocessing	Images are processed for training	P

Table 6.2 :Test Case for Training

Test Case Id	Test Scenario	Test Case	Pre Condition	Test Steps	Test Data	Expected	Post Condition	Actual Result	Test Status (P/F)
2	Load Yolov4 Weapon Detection Model	Training	Avail-ability of Dataset	Implement YOLOV4 algorithm	dataset	No Error	load Yolov4 model and then calculate its prediction accuracy	Accuracy is predicted	P

Table 6.3 :Test Case for Upload Image

Test Case Id	Test Scenario	Test Case	Pre Condition	Test Steps	Test Data	Expected	Post Condition	Actual Result	Test Status(P/F)
3	Upload Image	Prediction	Data Preprocessing	will upload test image	Dataset	No Error	will upload test image	will upload test image	P

Table 6.4 :Test Case for Detect Weapon from Image

Test Case Id	Test Scenario	Test Case	Pre Condition	Test Steps	Test Data	Expected	Post Condition	Actual Result	Test Status(P/F)
4	Detect Weapon from Image	Prediction	Data Preprocessing	Detect weapon using model	Dataset	No Error	apply Yolov4 model on loaded image to detect weapon	Weapon Detected	P

Table 6.5 :Test Case for Detect Weapon from Video

Test Case Id	Test Scenario	Test Case	Pre Condition	Test Steps	Test Data	Expected	Post Condition	Actual Result	Test Status(P/F)
5	Detect Weapon from Video	Prediction	Data Preprocessing	Detect weapon using Yolov4 model	Dataset	No Error	upload video and analyse each frame in the video to detect weapon	Weapon Detected	P

Table 6.6 :Test Case for Weapon Detection Training Accuracy-Loss Graph

Test Case Id	Test Scenario	Test Case	Pre Condition	Test Steps	Test Data	Expected	Post Condition	Actual Result	Test Status(P/F)
6	Accuracy-Loss Graph	Prediction	Data Preprocessing	will upload test image	Dataset	No Error	plot YoloV4 training accuracy and loss graph	Accuracy	P

7.RESULTS

7.1. Actual Results of the work

To run project double click on ‘run.bat’ file to get below screen

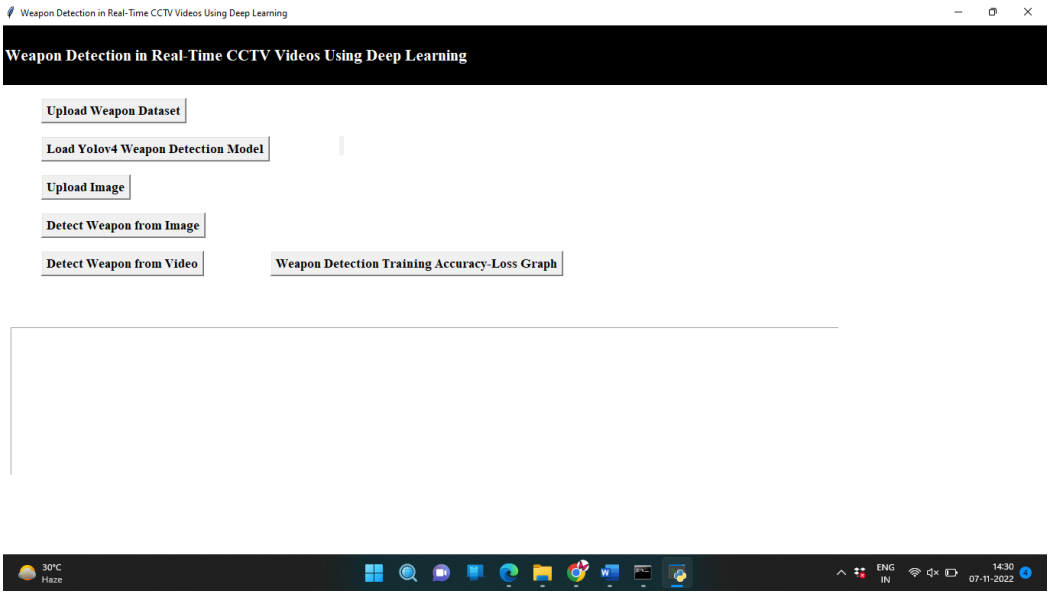


Fig7.1: Home Page

In above screen click on ‘Upload Weapon Dataset’ button to upload dataset and get below output

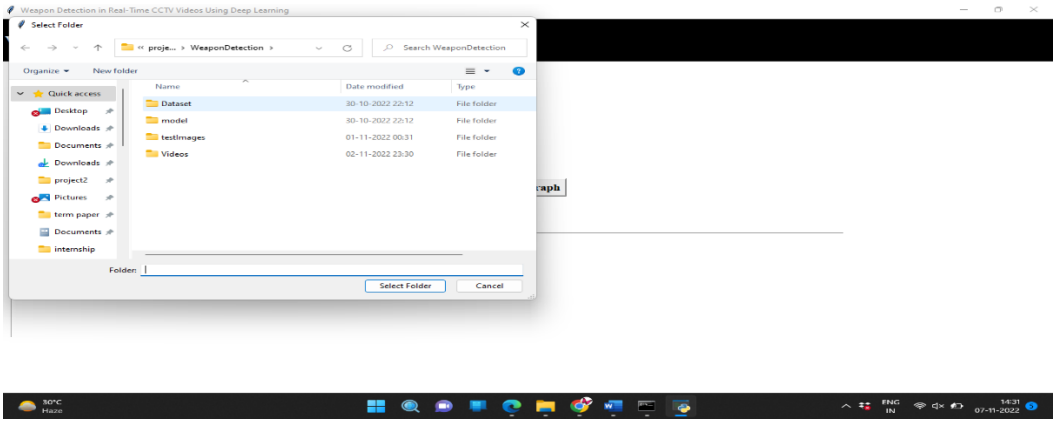


Fig7.2: Weapon Dataset Upload

In above screen selecting and uploading ‘Dataset’ folder and then click on ‘Select Folder’ button to load dataset and get below output.

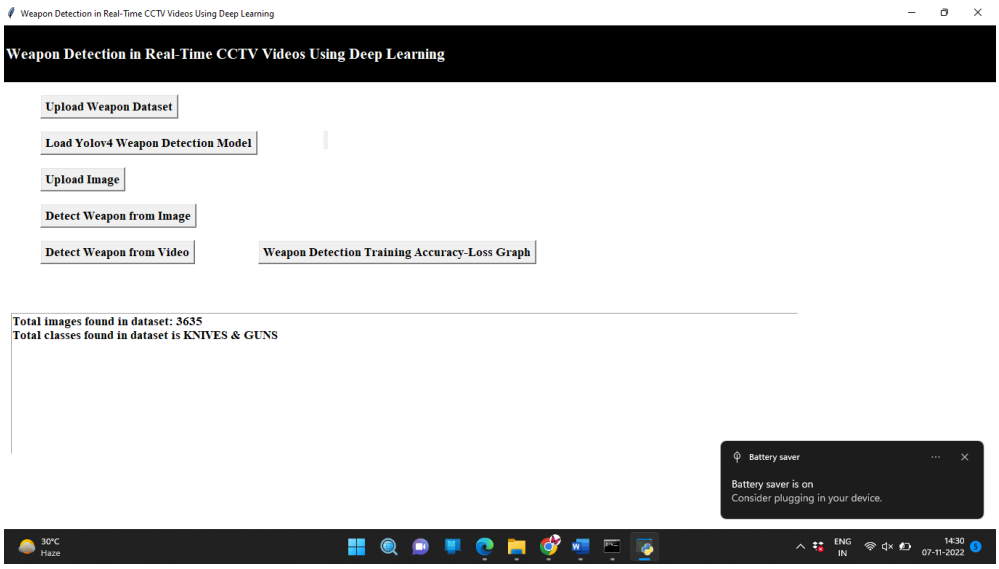


Fig7.3: Upload Dataset

In above screen we can see dataset loaded and dataset contains 2 classes called Guns and Knives and dataset contains total 3635 images and now close above image and then click on ‘Load Yolov4 Weapon Detection Model’ button to load YoloV4 model and calculate accuracy

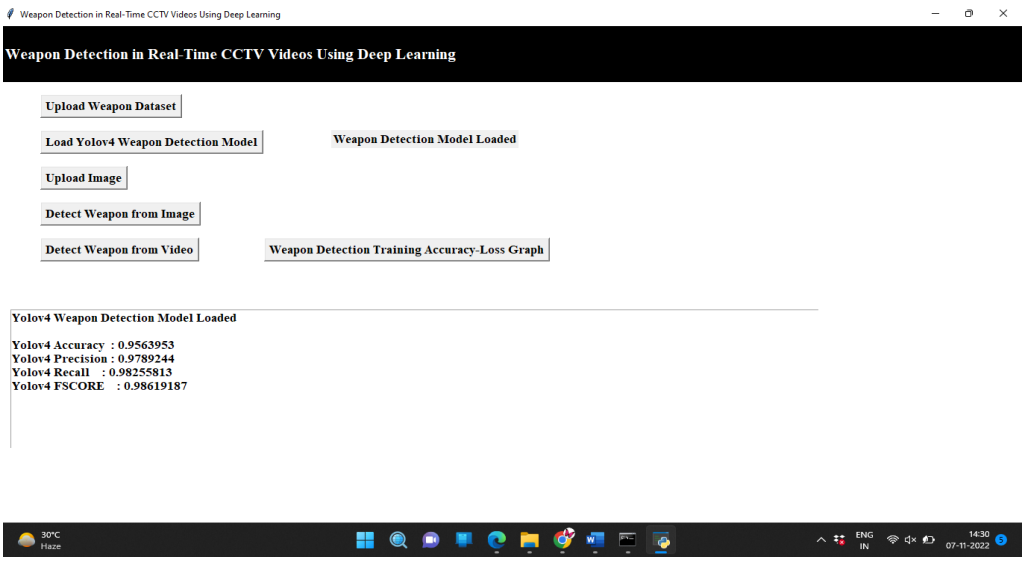


Fig7.4: Load YOLOV4 model

In above screen YoloV4 model loaded and we got its prediction accuracy as 95% and now model is loaded and now click on ‘Upload Image’ button to upload test image like below screen

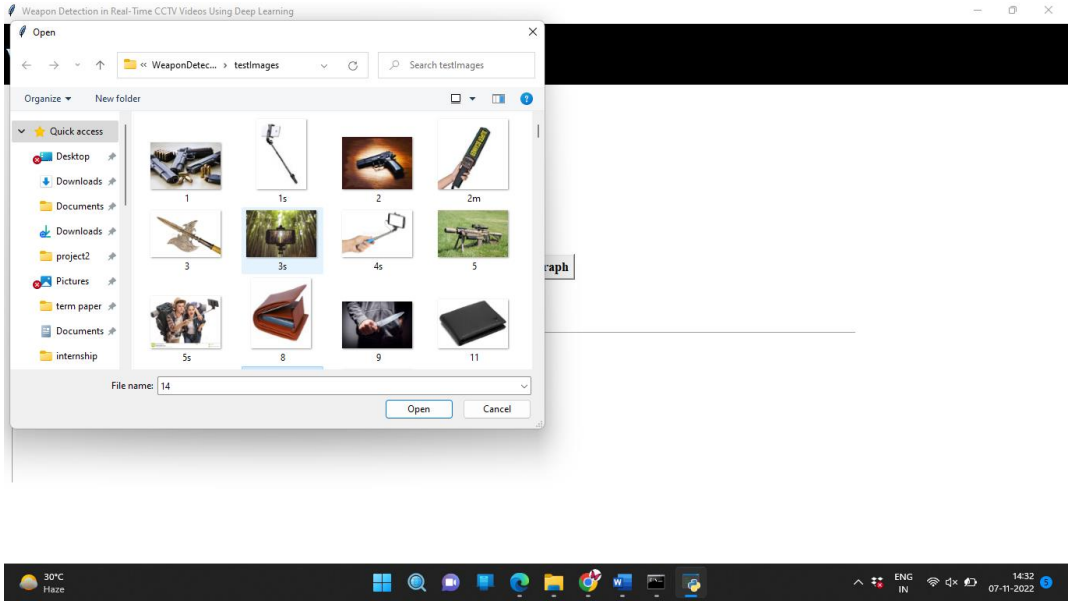


Fig7.5: Upload Image

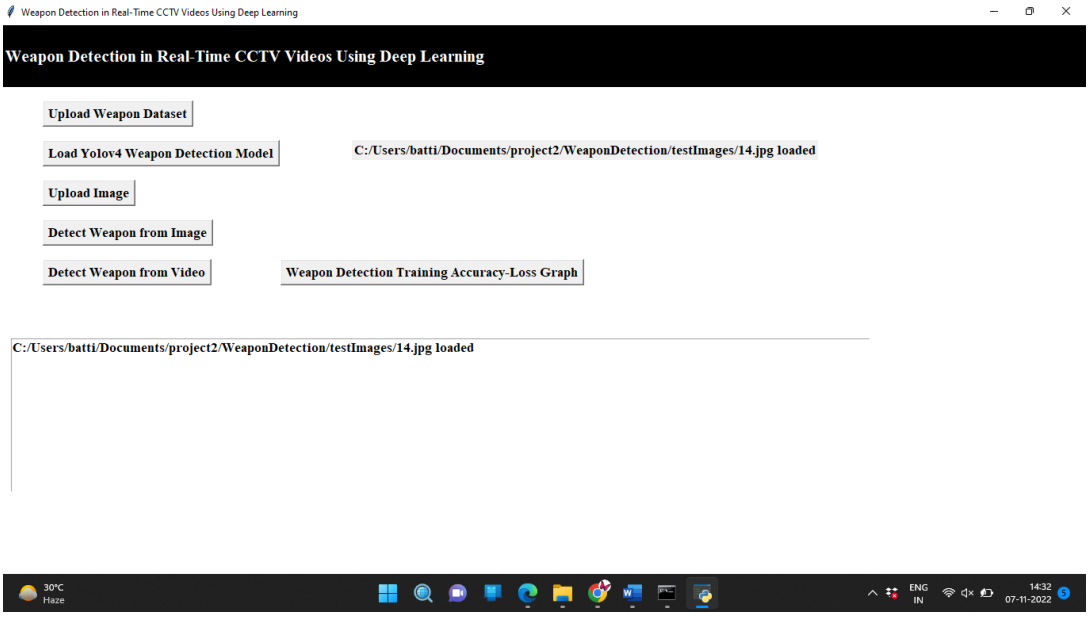


Fig7.6: Image Uploaded

In above screen selecting and uploading ‘14.jpg’ image and then click on ‘Open’ button to load image and then click on ‘Detect Weapon from Image’ button to detect weapon and get below output

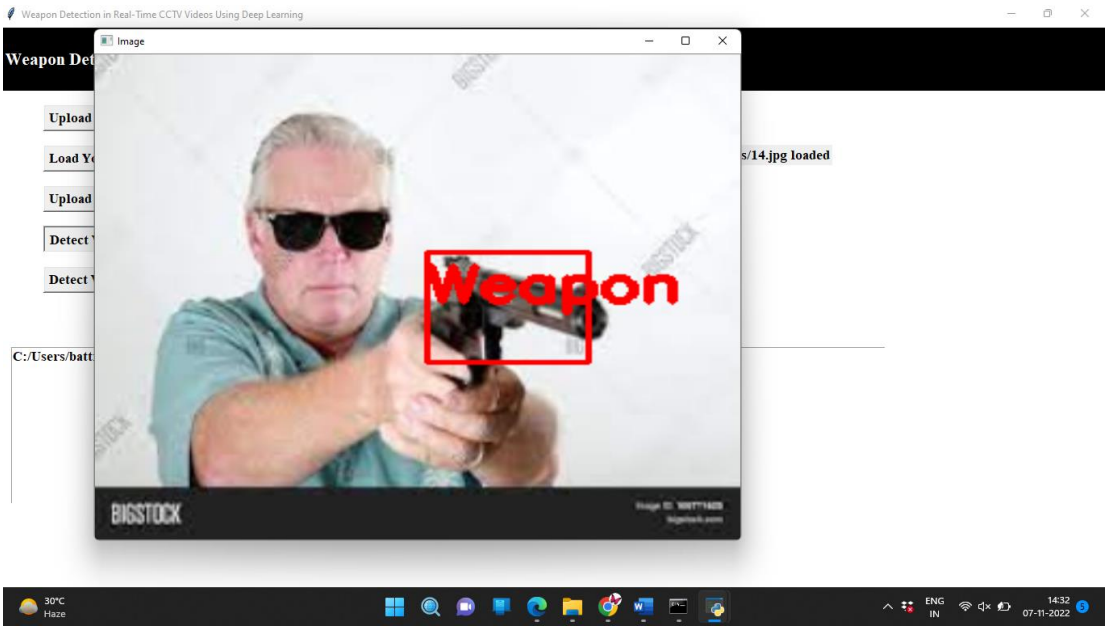


Fig7.7: Weapon Detection

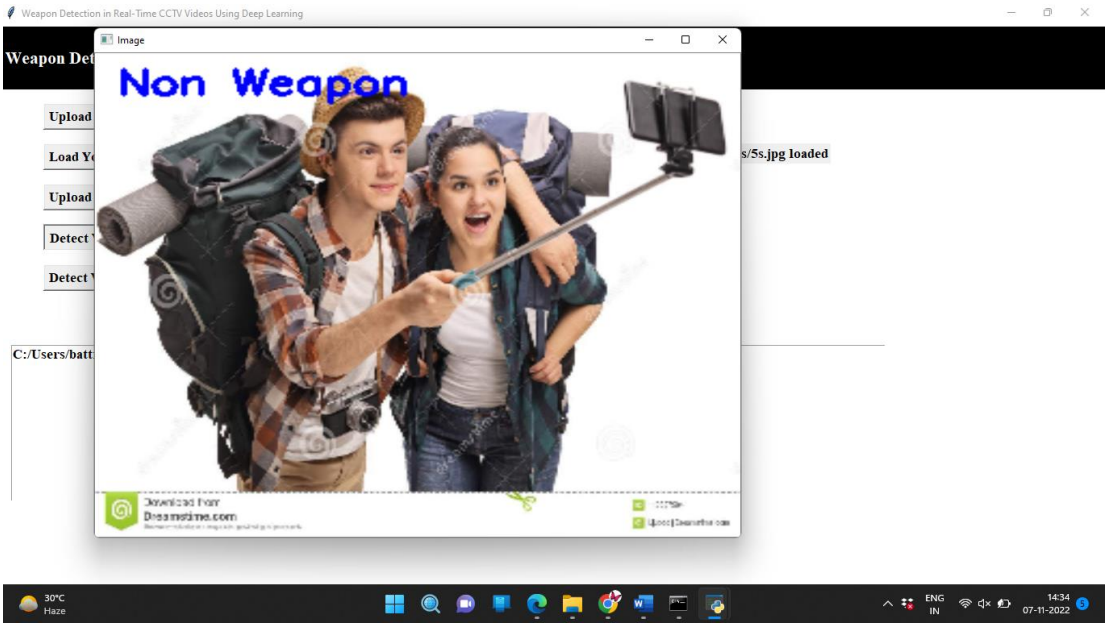


Fig7.8: Non-Weapon Detection

In above screen we got bounding box with label as ‘Weapon’ detect and will get beep sound also and similarly you can upload any image and get detection.

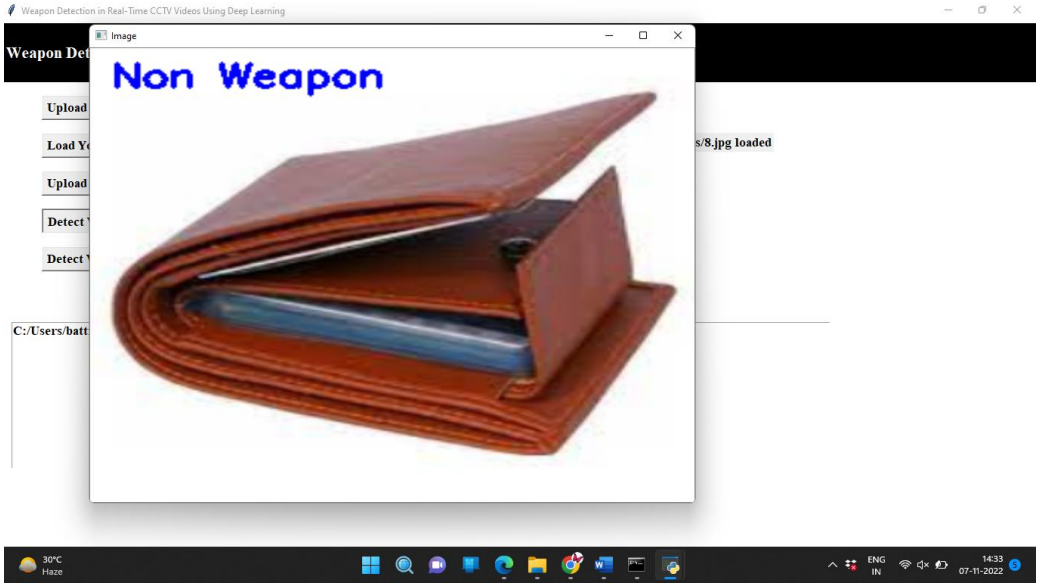


Fig7.9: Weapon Detection

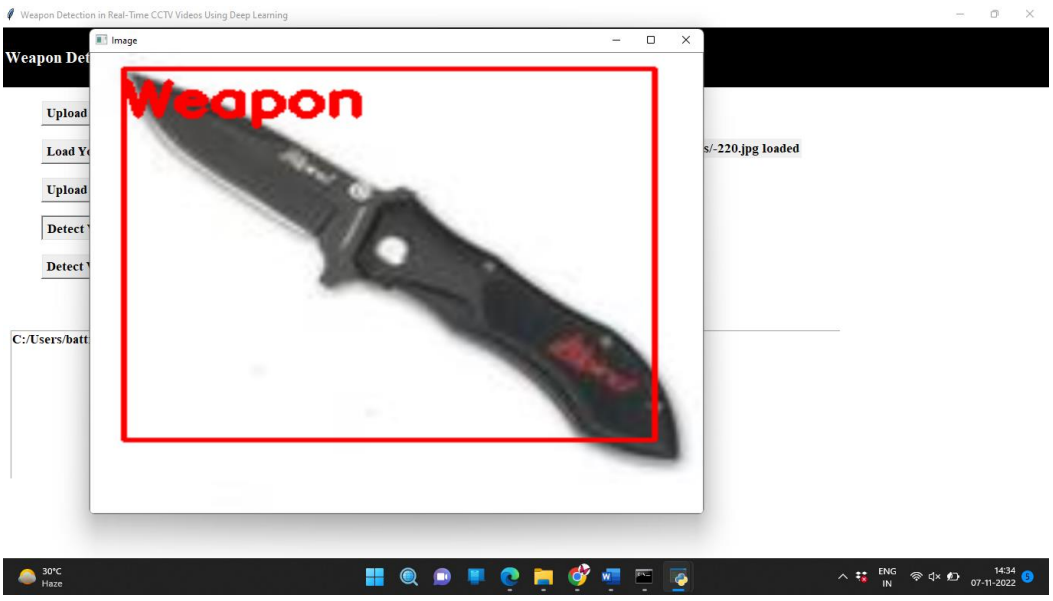


Fig7.10: Non-Weapon Detection

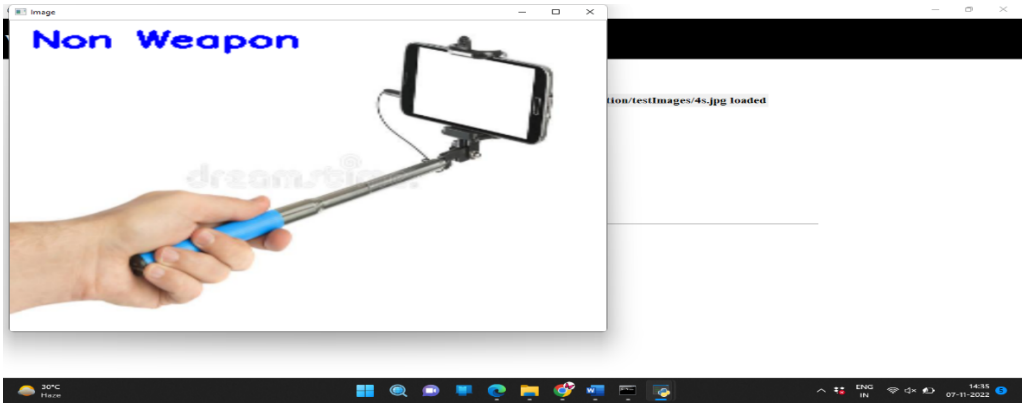


Fig7.11: Weapon Detection

In above image also weapon detected and now close above image and then click on 'Detect Weapon from Video' button to upload video and detect weapon from it

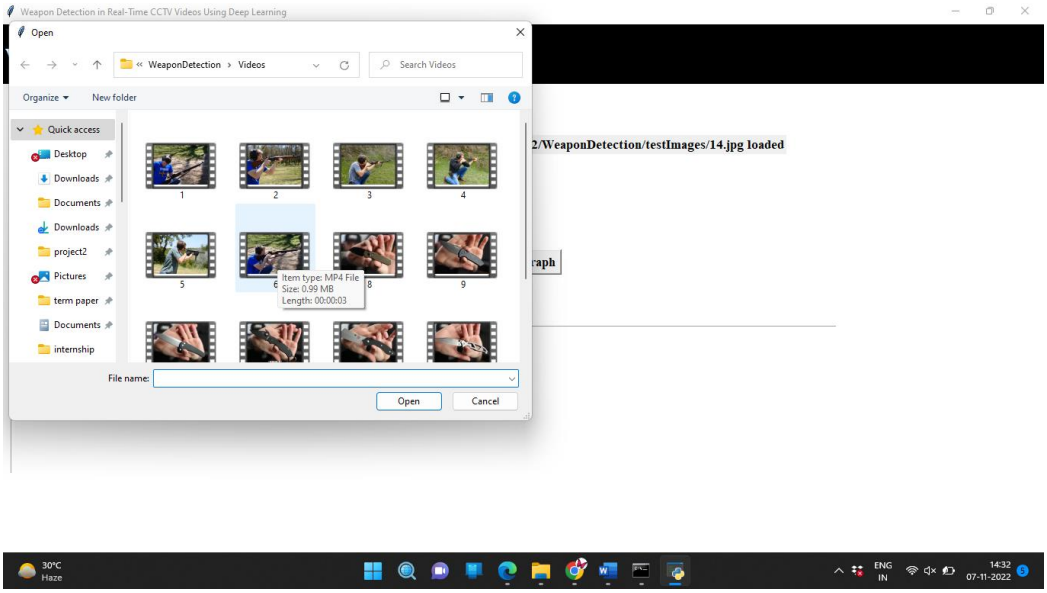


Fig7.12: Upload Video

In above screen selecting and uploading video file and then will get below output and this video will play slowly due to processing and if your system speed good then it will play faster

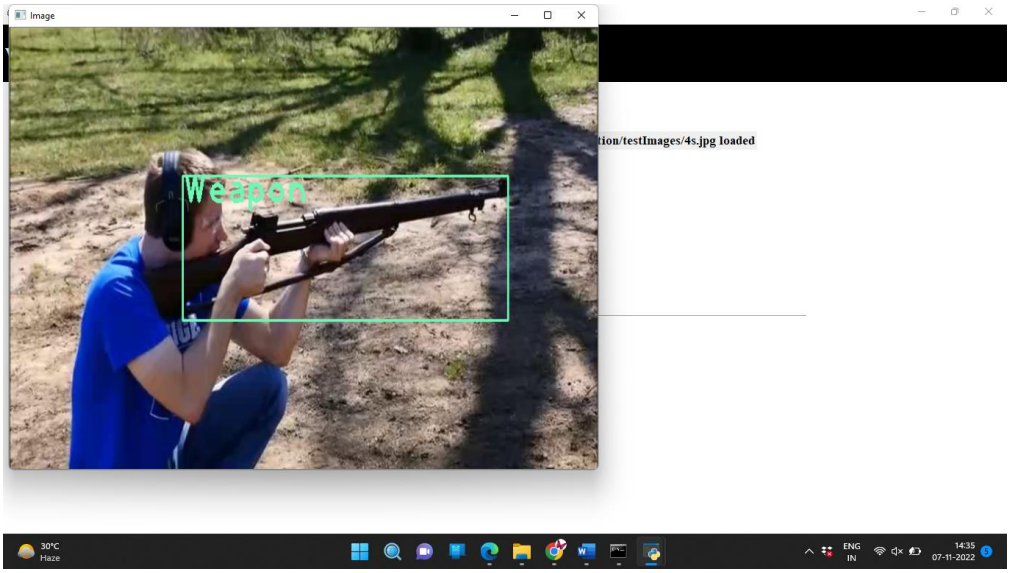


Fig7.13: Weapon Detection in Video

In above screen from video also model is detecting weapon and now close above video and then click on 'Weapon Detection Training Accuracy-Loss Graph' button to get Yolov4 training accuracy and loss graph

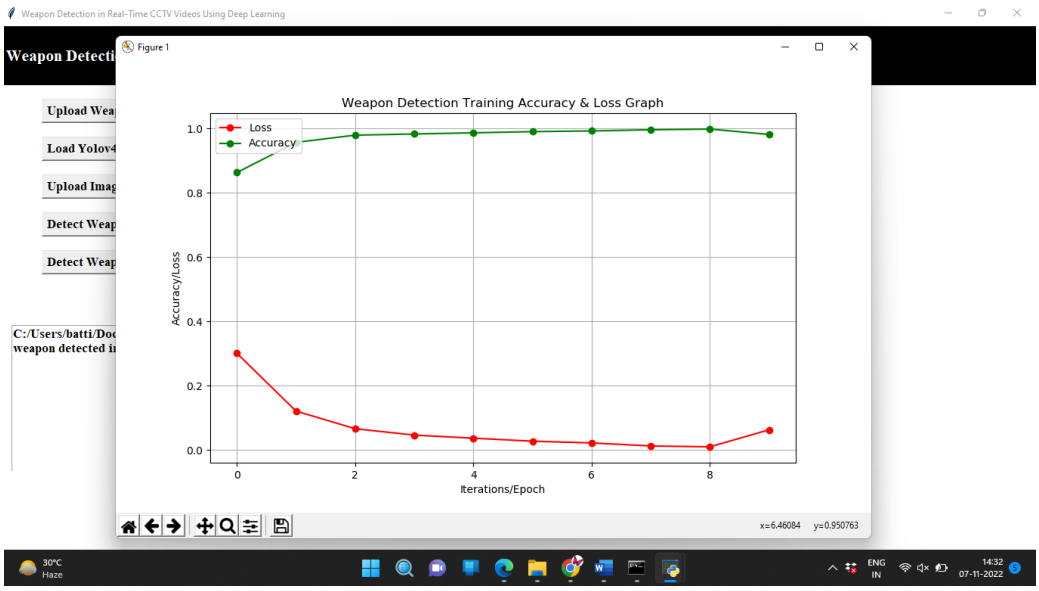


Fig7.14: Weapon Detection Training Accuracy-Loss Graph

In above graph x-axis represents training epoch and y-axis represents accuracy and loss values and green line represents accuracy and red line represents loss values and we can see with each increasing epoch accuracy got increase and reached closer to 1 and loss get decrease and reached closer to 0. Any model with increasing accuracy and decreasing loss consider as best model

8.CONCLUSION

For both monitoring and control purposes, this work has presented a novel automatic weapon detection system in realtime. This work will indeed help in improving the security, law and order situation for the betterment and safety of humanity, especially for the countries who had suffered a lot with these kind of violent activities. This will bring a positive impact on the economy by attracting investors and tourists, as security and safety are their primary needs. We have focused on detecting the weapon in live CCTV streams and at the same time reduced the false negatives and positives. To achieve high precision and recall we constructed a new training database for the real-time scenario, then trained, and evaluated it on the latest state-of-the-art deep learning models using two approaches, i.e. sliding window/classification and region proposal/object detection. Different algorithms were investigated to get good precision and recall.

6. REFRENCES

[1] (2019). Christchurch Mosque Shootzings. Accessed: Jul. 10, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Christchurch_mosque_ shootings

[2] (2019). Global Study on Homicide. Accessed: Jul. 10, 2019. [Online]. Available: https://www.unodc.org/unodc/en/data-and-analysis/globalstudy-on-homicide.html

[3] W. Deisman, ‘‘CCTV: Literature review and bibliography,’’ in Research and Evaluation Branch, Community, Contract and Aboriginal Policing Services Directorate. Ottawa, ON, Canada: Royal Canadian Mounted, 2003.

[4] J. Ratcliffe, ‘‘Video surveillance of public places,’’ US Dept. Justice, Office Community Oriented Policing Services, Washington, DC, USA, Tech. Rep. 4, 2006.

[5] M. Grega, A. Matiolański, P. Guzik, and M. Leszczuk, ‘‘Automated detection of firearms and knives in a CCTV image,’’ Sensors, vol. 16, no. 1, p. 47, Jan. 2016.

[6] TechCrunch. (2019). China’s CCTV Surveillance Network Took Just 7 Minutes to Capture BBC Reporter. Accessed: Jul. 15, 2019. [Online]. Available: https://techcrunch.com/2017/12/13/china-cctv-bbc-reporter/

[7] N. Cohen, J. Gattuso, and K. MacLennan-Brown. CCTV Operational Requirements Manual 2009. St Albans, U.K.: Home Office Scientific Development Branch, 2009.

[8] G. Flitton, T. P. Breckon, and N. Megherbi, ‘‘A comparison of 3D interest point descriptors with application to airport baggage object detection in complex CT imagery,’’ Pattern Recognit., vol. 46, no. 9, pp. 2420–2436, Sep. 2013.

[9] R. Gesick, C. Saritac, and C.-C. Hung, ‘‘Automatic image analysis process for the detection of concealed weapons,’’ in Proc. 5th Annu. Workshop Cyber Secur. Inf. Intell. Res. Cyber Secur. Inf. Intell. Challenges Strategies (CSIIRW), 2009, p. 20.

[10] R. K. Tiwari and G. K. Verma, ‘‘A computer vision based framework for visual gun detection using Harris interest point detector,’’ Procedia Comput. Sci., vol. 54, pp. 703–712, Aug. 2015.

[11] R. K. Tiwari and G. K. Verma, ‘‘A computer vision based framework for visual gun detection using SURF,’’ in Proc. Int. Conf. Electr., Electron., Signals, Commun. Optim. (EESCO), Jan. 2015, pp. 1–5.

[12] Z. Xiao, X. Lu, J. Yan, L. Wu, and L. Ren, ‘‘Automatic detection of concealed pistols using passive millimeter wave imaging,’’ in Proc. IEEE Int. Conf. Imag. Syst. Techn. (IST), Sep. 2015, pp. 1–4.

[13] D. M. Sheen, D. L. Mcmakin, and T. E. Hall, ‘‘Three-dimensional millimeter-wave imaging for concealed weapon detection,’’ IEEE Trans. Microw. Theory Techn., vol. 49, no. 9, pp. 1581–1592, Sep. 2001.

[14] Z. Xue, R. S. Blum, and Y. Li, ‘‘Fusion of visual and IR images for concealed weapon detection,’’ in Proc. 5th Int. Conf. Inf. Fusion, vol. 2, Jul. 2002, pp. 1198–1205.

[15] R. Blum, Z. Xue, Z. Liu, and D. S. Forsyth, ‘‘Multisensor concealed weapon detection by using a multiresolution mosaic approach,’’ in Proc. IEEE 60th Veh. Technol. Conf. (VTC-Fall), vol. 7, Sep. 2004, pp. 4597–4601.

[16] E. M. Upadhyay and N. K. Rana, ‘‘Exposure fusion for concealed weapon detection,’’ in Proc. 2nd Int. Conf. Devices, Circuits Syst. (ICDCS), Mar. 2014, pp. 1–6.

[17] R. Maher, ‘‘Modeling and signal processing of acoustic gunshot recordings,’’ in Proc. IEEE 12th Digit. Signal Process. Workshop 4th IEEE Signal Process. Educ. Workshop, Sep. 2006, pp. 257–261.

[18] A. Chacon-Rodriguez, P. Julian, L. Castro, P. Alvarado, and N. Hernandez, ‘‘Evaluation of gunshot detection algorithms,’’ IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 58, no. 2, pp. 363–373, Feb. 2011.

[19] (2019). From Edison to Internet: A History of Video Surveillance. Accessed: Jun. 13, 2019. [Online]. Available: https://www. business2community.com/tech-gadgets/from-edison-to-internet-a history-of-video-surveillance-0578308

[20] (2019). Infographic: History of Video Surveillance—IFSEC Global | Security and Fire News and Resources. Accessed: Sep. 15, 2019. [Online]. Available: https://www.ifsecglobal.com/video-surveillance/infographichistory-of-video-surveillance/

[21] W. Hu, T. Tan, L. Wang, and S. Maybank, ‘‘A survey on visual surveillance of object motion and behaviors,’’ IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., vol. 34, no. 3, pp. 334–352, Aug. 2004.

[22] A. C. Sankaranarayanan, A. Veeraraghavan, and R. Chellappa, ‘‘Object detection, tracking and recognition for multiple smart cameras,’’ Proc. IEEE, vol. 96, no. 10, pp. 1606–1624, Oct. 2008.

[23] S. Zhang, C. Wang, S.-C. Chan, X. Wei, and C.-H. Ho, ‘‘New object detection, tracking, and recognition approaches for video surveillance over camera network,’’ IEEE Sensors J., vol. 15, no. 5, pp. 2679–2691, May 2015.

[24] J. C. Nascimento and J. S. Marques, ‘‘Performance evaluation of object detection algorithms for video surveillance,’’ IEEE Trans. Multimedia, vol. 8, no. 4, pp. 761–774, Aug. 2006.

[25] N. Dalal and B. Triggs, ‘‘Histograms of oriented gradients for human detection,’’ Tech. Rep., 2005.

[26] C. Anagnostopoulos, I. Anagnostopoulos, G. Tsekouras, G. Kouzas, V. Loumos, and E. Kayafas, ‘‘Using sliding concentric windows for license plate segmentation and processing,’’ in Proc. IEEE Workshop Signal Process. Syst. Design Implement., Nov. 2005, pp. 337–342.

[27] M. Grega, S. Lach, and R. Sieradzki, ‘‘Automated recognition of firearms in surveillance video,’’ in Proc. IEEE Int. Multi-Disciplinary Conf. Cognit. Methods Situation Awareness Decis. Support (CogSIMA), Feb. 2013, pp. 45–50.

[28] I. Darker, A. Gale, L. Ward, and A. Blechko, ‘‘Can CCTV reliably detect gun crime?’’ in Proc. 41st Annu. IEEE Int. Carnahan Conf. Secur. Technol., Oct. 2007, pp. 264–271.

[29] I. T. Darker, A. G. Gale, and A. Blechko, ‘‘CCTV as an automated sensor for firearms detection: Human-derived performance as a precursor to automatic recognition,’’ Proc. SPIE, vol. 7112, Oct. 2008, Art. no. 71120V.

[30] I. T. Darker, P. Kuo, M. Y. Yang, A. Blechko, C. Grecos, D. Makris, J.-C. Nebel, and A. Gale, ‘‘Automation of the CCTV-mediated detection of individuals illegally carrying firearms: Combining psychological and technological approaches,’’ Proc. SPIE, vol. 7341, Apr. 2009, Art. no. 73410P.

[31] R. Al-Rfou et al., ‘‘Theano: A Python framework for fast computation of mathematical expressions,’’ 2016, arXiv:1605.02688. [Online]. Available: http://arxiv.org/abs/1605.02688

[32] F. Chollet. (2019). Fchollet. Accessed: Apr. 10, 2019. [Online]. Available: https://github.com/fchollet

[33] K. He, X. Zhang, S. Ren, and J. Sun, ‘‘Deep residual learning for image recognition,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[34] (2016). Weapon Detection in Surveillance Cameras Images. Accessed: Feb. 13, 2021. [Online]. Available: http://www.divaportal.org/smash/record.jsf?pid=diva2%3A1054902&dswid=-1974

[35] M. Nakib, R. T. Khan, M. S. Hasan, and J. Uddin, ‘‘Crime scene prediction by detecting threatening objects using convolutional neural network,’’ in Proc. Int. Conf. Comput., Commun., Chem., Mater. Electron. Eng. (IC4ME2), Feb. 2018, pp. 1–4.

[36] G. K. Verma and A. Dhillon, ‘‘A handheld gun detection using faster RCNN deep learning,’’ in Proc. 7th Int. Conf. Comput. Commun. Technol. (ICCCT), 2017, pp. 84–88.

[37] R. Olmos, S. Tabik, and F. Herrera, ‘‘Automatic handgun detection alarm in videos using deep learning,’’ Neurocomputing, vol. 275, pp. 66–72, Jan. 2018.

[38] J. Iqbal, M. A. Munir, A. Mahmood, A. Rafaqat Ali, and M. Ali, ‘‘Orientation aware object detection with application to firearms,’’ 2019, arXiv:1904.10032. [Online]. Available: https://arxiv.org/abs/1904.10032

[39] J. L. S. González, C. Zaccaro, J. A. Álvarez-García, L. M. S. Morillo, and F. S. Caparrini, ‘‘Real-time gun detection in CCTV: An open problem,’’ Neural Netw., vol. 132, pp. 297–308, Dec. 2020.

[40] (2017). Convolutional Neural Networks. Accessed: Aug. 15, 2018. [Online]. Available: http://cs231n.github.io/convolutional-networks/

[41] K. Simonyan and A. Zisserman, ‘‘Very deep convolutional networks for large-scale image recognition,’’ 2014, arXiv:1409.1556. [Online]. Available: http://arxiv.org/abs/1409.1556

[42] (2019). VGG16—Convolutional Network for Classification and Detection. Accessed: Dec. 19, 2018. [Online]. Available: https://neurohive. io/en/popular-networks/vgg16/

[43] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, ‘‘Rethinking the inception architecture for computer vision,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2818–2826.

[44] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, ‘‘Inception-v4, inception-resnet and the impact of residual connections on learning,’’ in Proc. 31st AAAI Conf. Artif. Intell., 2017, pp. 1–7.

[45] Medium. (2019). A Simple Guide to the Versions of the Inception Network. Accessed: Jul. 27, 2019. [Online]. Available: https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the inception-network-7fc52b863202

[46] D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, ‘‘SSD: Single shot multibox detector,’’ in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2016, pp. 21–37.

[47] Medium. (2019). Understanding SSD MultiBox—Real-Time Object Detection in Deep Learning. Accessed: Aug. 19, 2019. [Online]. Available: https://towardsdatascience.com/understanding-ssd-multibox-real-time-object-detection-in-deep-learning-495ef744fab

[48] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, ‘‘MobileNets: Efficient convolutional neural networks for mobile vision applications,’’ 2017, arXiv:1704.04861. [Online]. Available: http://arxiv.org/abs/1704.04861

[49] J. Redmon and A. Farhadi, ‘‘YOLOv3: An incremental improvement,’’ 2018, arXiv:1804.02767. [Online]. Available: http://arxiv.org/abs/ 1804.02767

[50] S. Ren, K. He, R. Girshick, and J. Sun, ‘‘Faster R-CNN: Towards real-time object detection with region proposal networks,’’ in Proc. Adv. Neural Inf. Process. Syst., 2015. pp. 91–99.

[51] (2019). Faster R-CNN Explained. Accessed: Aug. 25, 2019. [Online]. Available: https://medium. com/@smallfishbigsea/faster-r-cnn-explained-864d4fb7e3f

[52] Medium. (2019). Faster RCNN Object detection. Accessed: Aug. 27, 2019. [Online]. Available: https://towardsdatascience.com/faster-rcnn-object-detection-f865e5ed7fc4

[53] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, ‘‘YOLOv4: Optimal speed and accuracy of object detection,’’ 2020, arXiv:2004.10934. [Online]. Available: http://arxiv.org/abs/2004.10934

[54] GeeksforGeeks. (2020). Object Detection Vs Object Recognition Vs Image Segmentation. Accessed: Dec. 28, 2020. [Online]. Available: https://www.geeksforgeeks.org/object-detection-vs-object-recognition-vs-image-segmentation/

[55] (2019). ImageNet. Accessed: Jun. 5, 2019. [Online]. Available: http:// www.image-net.org/

[56] J. O. Laguna, A. G. Olaya, and D. Borrajo, ‘‘A dynamic sliding window approach for activity recognition,’’ in Proc. Int. Conf. User Modeling, Adaptation, Personalization. Berlin, Germany: Springer, 2011, pp. 219–230.

[57] J. Hosang, R. Benenson, P. Dollar, and B. Schiele, ‘‘What makes for effective detection proposals?’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 4, pp. 814–830, Apr. 2016.

[58] R. Girshick, J. Donahue, T. Darrell, and J. Malik, ‘‘Rich feature hierarchies for accurate object detection and semantic segmentation,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 580–587.

[59] A. Consulting. (2019). Selective Search for Object Detection (C++ / Python) | Learn OpenCV. Accessed: May 25, 2019. [Online]. Available: https://www.learnopencv.com/selective-search-for-object-detection-cpp-python/

[60] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, ‘‘Gradient-based learning applied to document recognition,’’ Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[61] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, ‘‘Data preprocessing for supervised leaning,’’ Int. J. Comput. Sci., vol. 1, no. 2, pp. 111–117, 2006.