**WEAPON DETECTION IN REAL-TIME CCTV VIDEOS USING DEEP LEARNING**

**ABSTRACT**

**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

|  |  |  |
| --- | --- | --- |
| S.NO | CONTENT | PGNO |
| 1 | Introduction   * 1. Software requirements   2. Hardware requirements |  |
| 2 | Feasibility study  2.1 Economic feasibility  2.2 Technical feasibility  2.3 Social feasibility |  |
| 3 | Literature survey |  |
| 4 | System analysis  4.1 Existing system  4.1.1 Disadvantages of existing system  4.2 Proposed system  4.2.1 Advantages of proposed system  4.3 Functional requirements  4.4 Non-Functional requirements |  |
| 5 | System design  5.1 System architecture  5.2 UML diagrams |  |
| 6 | Implementation  6.1 Modules  6.2 Sample code |  |
| 7 | Software environment |  |
| 8 | System testing  8.1 Testing strategies  8.2 Test cases |  |
| 9 | Screens |  |
| 10 | Conclusion |  |
| 11 | References |  |

LIST OF FIGURE

|  |  |  |
| --- | --- | --- |
| FIG.NO | FIG.NAME | PG.NO |
| 5.1.1 | System architecture |  |
| 5.1.2 | Dataflow diagram |  |
| 5.1.3 | Usecase diagram |  |
| 5.2.1 | Class diagram |  |
| 5.2.2 | Activity diagram |  |
| 5.2.3 | Sequence diagram |  |
| 5.2.4 | Collaboration diagram |  |
| 5.2.5 | Component diagram |  |
| 5.2.6 | Deployment diagram |  |

**INTRODUCTION**

**1.INTRODUCTION**

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. High incidents were recorded in past few years with the use of harmful weapons in public areas. Starting with the past year’s attacks on a couple of Mosques in New Zealand, on March 15, 2019 at 1:40 pm, the attacker attacks the Christchurch AL-Noor Mosque during a Friday prayer killing almost 44 innocent and unarmed worshippers. On the same day just after 15 minutes at 1:55 PM, another attack happened killing seven more civilians [1]. Active shooter incidents had also occurred in USA and then in Europe. The most significant cases were those at Columbine High School (USA, 37 victims), Andreas Broeivik’s assault on Uotya Island (Norway, 179 victims) or the Charlie Hebdo newspaper attack killing 23. According to stats provided by the UNODC, among 0.1 Million people of a country, the crimes involving guns are very high i-e. 1.6 in Belgium, United States having 4.7 and Mexico with a number of 21.5.

**1.1 SOFTWARE REQUIREMENTS**

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

* **Python idel 3.7 version (or)**
* **Anaconda 3.7 ( or)**
* **Jupiter (or)**
* **Google colab**

**1.2 HARDWARE REQUIREMENTS**

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

* **Operating system : windows, linux**
* **Processor : minimum intel i3**
* **Ram : minimum 4 gb**
* **Hard disk : minimum 250gb**

**FEASIBILITY STUDY**

**2. FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**2.1 ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### **2.2 TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**2.3 SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**LITERATURE SURVEY**

**3.LITERATURE SURVEY**

**3.1 Video surveillance of public places**

**https://www.researchgate.net/publication/252218884\_Video\_Surveillance\_of\_Public\_Places**

Designed for police officers of any rank and assignment, the Problem-Specific Guides summarize knowledge about how police can analyze and reduce particular crime problems in local communities. This guide begins with an analysis of the benefits and problems associated with the use of CCTV systems for public surveillance purposes, which include the unintended consequences of displacement of crime, increased citizen fear of crime, and increased reporting of certain types of crimes. Benefits are, ironically, a reduced fear of crime and the ability to quickly recognize and respond to a medical emergency. Evaluation findings of CCTV programs are presented which generally suggest that CCTV is better at combating property crimes than violent or public order crimes; CCTV works best in small, well-defined areas; and achieving statistically significant crime reductions with CCTV can be difficult. Implementation considerations are discussed and include a series of questions to guide police departments in their decision on whether to implement a system. Questions direct police departments to consider whether CCTV is the best option, where the cameras will be located, who will operate the system, and who and what should be watched. Police departments should also evaluate whether they have the human and budgetary capacity to run an effective CCTV program. Managing the public concern should also be a consideration of police departments and generally involves issues related to privacy and constitutionality. Finally, the next generation technology is briefly described as low-level x-ray imaging and facial recognition systems. Appendixes, endnotes, references, recommended readings.

**3.2 Automated detection of firearms and knives in a CCTV image**

**https://www.researchgate.net/publication/288872331\_Automated\_Detection\_of\_Firearms\_and\_Knives\_in\_a\_CCTV\_Image**

Closed circuit television systems (CCTV) are becoming more and more popular and are being deployed in many offices, housing estates and in most public spaces. Monitoring systems have been implemented in many European and American cities. This makes for an enormous load for the CCTV operators, as the number of camera views a single operator can monitor is limited by human factors. In this paper, we focus on the task of automated detection and recognition of dangerous situations for CCTV systems. We propose algorithms that are able to alert the human operator when a firearm or knife is visible in the image. We have focused on limiting the number of false alarms in order to allow for a real-life application of the system. The specificity and sensitivity of the knife detection are significantly better than others published recently. We have also managed to propose a version of a firearm detection algorithm that offers a near-zero rate of false alarms. We have shown that it is possible to create a system that is capable of an early warning in a dangerous situation, which may lead to faster and more effective response times and a reduction in the number of potential victims.

**3.3 A comparison of 3D interest point descriptors with application to airport baggage object detection in complex CT imagery**

**https://www.sciencedirect.com/science/article/abs/pii/S0031320313000848**

We present an experimental comparison of 3D feature descriptors with application to threat detection in Computed Tomography (CT) airport baggage imagery. The detectors range in complexity from a basic local density descriptor, through local region histograms and three-dimensional (3D) extensions to both to the RIFT descriptor and the seminal SIFT feature descriptor. We show that, in the complex CT imagery domain containing a high degree of noise and imaging artefacts, a specific instance object recognition system using simpler descriptors appears to outperform a more complex RIFT/SIFT solution. Recognition rates in excess of 95% are demonstrated with minimal false-positive rates for a set of exemplar 3D objects.

**3.4 Automatic image analysis process for the detection of concealed weapons**

**https://www.semanticscholar.org/paper/Automatic-image-analysis-process-for-the-detection-Gesick-Saritac/1e7a2d629f7d6efdfd28e16865d0c92f10badf8a**

The goal of this research is to develop a process, using current imaging hardware and without human intervention, that provides an accurate and timely detection alert of a concealed weapon and its location in the image of the luggage. There are several processes in existence that are able to highlight or otherwise outline a concealed weapon in baggage but so far those processes still require a highly trained operator to observe the resulting image and draw the correct conclusions. We attempted three different approaches in this project. The first approach uses edge detection combined with pattern matching to determine the existence of a concealed pistol. Rather than use the whole body of the weapon which varies significantly, the trigger guard was used since it is fairly consistent in dimensions. While the processes were reliable in detecting a pistol's presence, on any but the simplest of images, the computational time was excessive and a substantial number of false positives were generated. The second approach employed Daubechie wavelet transforms but the results have so far been inconclusive. A third approach involving an algorithm based on the scale invariant feature transform (SIFT) is proposed.

**3.5 A computer vision based framework for visual gun detection using Harris interest point detector**

**https://www.researchgate.net/publication/281643025\_A\_Computer\_Vision\_based\_Framework\_for\_Visual\_Gun\_Detection\_Using\_Harris\_Interest\_Point\_Detector**

oday's automatic visual surveillance is prime need for security and this paper presents first step in the direction of automatic visual gun detection. The objective of our paper is to develop a framework for visual gun detection for automatic surveillance. The proposed framework exploits the color based segmentation to eliminate unrelated object from an image using k-mean clustering algorithm. Harris interest point detector and Fast Retina Keypoint (FREAK) is used to locate the object (gun) in the segmented images. Our framework is robust enough in terms of scale, rotation, affine and occlusion. We have implemented and tested the system over sample images of gun, collected by us. We got promising performance of our system to detect a gun. Further, our system performs very well under different appearance of images. Thus our system is rotation, scale and shape invariant.

**SYSTEM ANALYSIS**

**4.SYSTEM ANALYSIS**

**4.1 EXISTING SYSTEM:**

Different approaches then used for weapon detection using sliding window and region proposal algorithms. HOG (Histogram of oriented Gradient) models were used to predict the objects in the frame. HOG significant work used lowlevel features, discriminative learning, and pictorial structure along with SVM. These algorithms were slow for real-time scenarios with 14s per image. Although these classifiers gave good accuracies, the slowness of the sliding window method was a big problem, especially for the realtime implementation purpose.

**4.1.1 DISADVANTAGES OF EXISTING SYSTEM:**

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

# 4.2 Proposed System:

This article presents an automatic detection and classification method of weapons for real-time scenario using state of the art deep learning models. For real-time implementation relating the problem question of this work ‘‘detecting weapons in real-time for potential robbers/terrorist using deep learning’’, detection and classification was done for pistol, revolver and other shot handheld weapons as in single class called pistol and related confusion objects such as cell phone, metal detector, wallet, selfie stick in not pistol class. A major reason behind this was our research done on weapons used in robbery cases and it further motivated us to choose pistol and revolver as our target object. We go through several CCTV captured robbery videos on YouTube and found that almost 95% of cases have pistol or revolver as the weapon used.

# 4.2.1 Advantages of proposed system:

1. Precision and recall count the most rather than accuracy when object detection is performed so these entire algorithms were tested in terms of them.
2. In all algorithms Yolov4 is giving more than 90% accuracy and FSCORE.

### **4.3 FUNCTIONAL REQUIREMENTS**

1.Data Collection

2.Data Preprocessing

3.Training And Testing

4.Modiling

5.Predicting

### **4.4 NON FUNCTIONAL REQUIREMENTS**

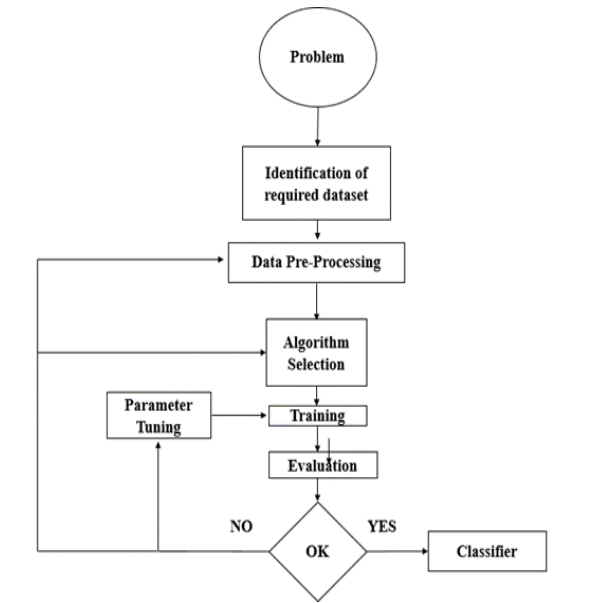
NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, *“how fast does the website load?”* Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non- functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users are > 10000. Description of non-functional requirements is just as critical as a functional requirement.

* Usability requirement
* Serviceability requirement
* Manageability requirement
* Recoverability requirement
* Security requirement
* Data Integrity requirement
* Capacity requirement
* Availability requirement
* Scalability requirement
* Interoperability requirement
* Reliability requirement
* Maintainability requirement
* Regulatory requirement
* Environmental requirement

**SYSTEM DESIGN**

**5. SYSTEM DESIGN**

**5.1 SYSTEM ARCHITECTURE:**

****

**Fig.5.1.1 System architecture**

**DATA FLOW DIAGRAM:**

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

Upload Weapon Dataset

**VERIFY**

**NO PROCESS**

**Yes NO**

Load Yolov4 Weapon Detection Model

Upload Image

Detect Weapon from Image

Detect Weapon from Video

Weapon Detection Training Accuracy-Loss Graph

**End process**

**Fig.5.1.3 Dataflow diagram**

**5.2 UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

**Use case diagram:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



**Fig.5.2.1 Usecase diagram**

**Class diagram:**

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.



**Fig.5.2.2 Class diagram**

**Activity diagram:**

The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.

Upload weapon dataset

Load YoloV4 weapon detection model

Upload image

Detect weapon from image

Detect weapon from video

Weapon detection training accuracy-loss graph

**Fig.5.2.3 Activity diagram**

**Sequence diagram:**

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".



**Fig.5.2.4 Sequence diagram**

**Collaboration diagram:**

A collaboration diagram groups together the interactions between different objects. The interactions are listed as numbered interactions that help to trace the sequence of the interactions. The collaboration diagram helps to identify all the possible interactions that each object has with other objects.

****

**Fig.5.2.5 Collaboration diagram**

**Component diagram:**

The component diagram represents the high-level parts that make up the system. This diagram depicts, at a high level, what components form part of the system and how they are interrelated. A component diagram depicts the components culled after the system has undergone the development or construction phase.



**Fig.5.2.6 Component diagram**

**Deployment diagram:**

The deployment diagram captures the configuration of the runtime elements of the application. This diagram is by far most useful when a system is built and ready to be deployed.

****

**Fig.5.2.7 Deployment diagram**

**IMPLEMENTATION**

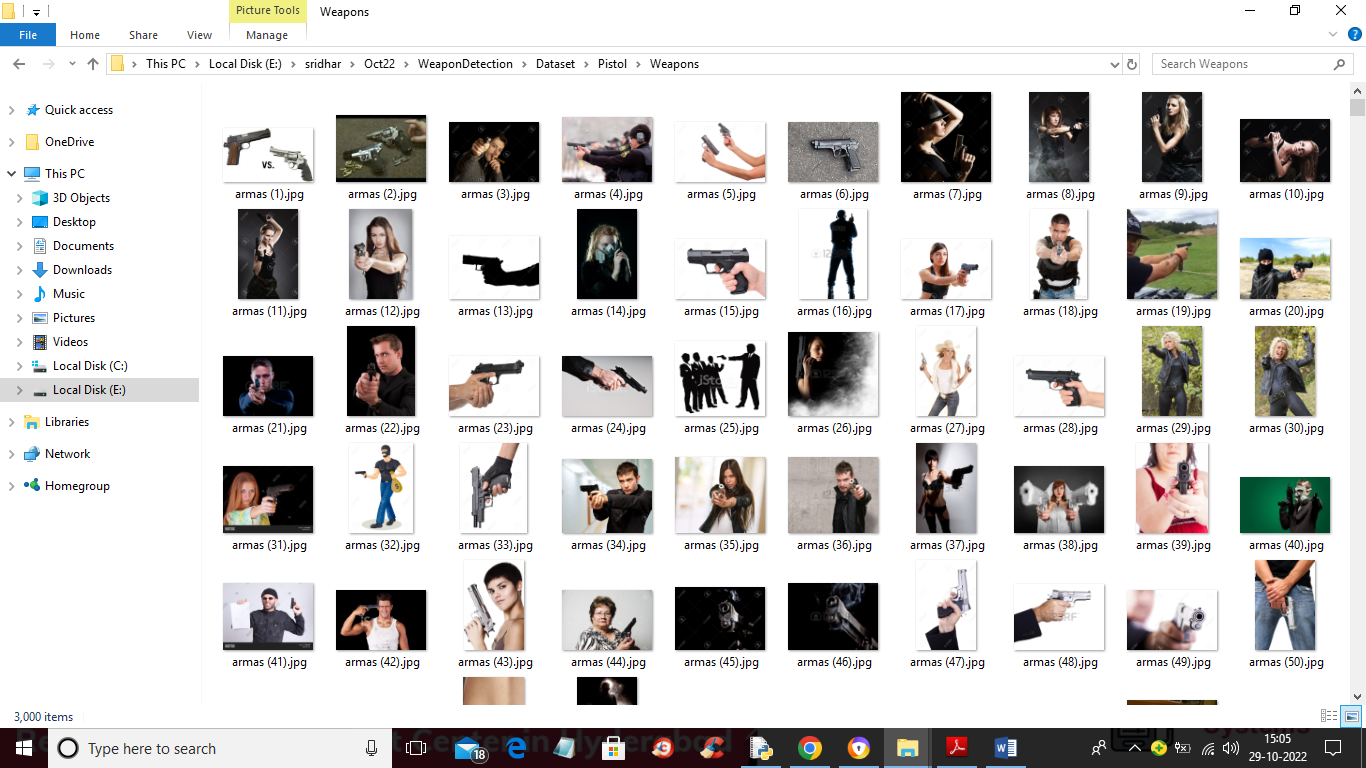
1. **IMPLEMENTATION**

In this paper author is experimenting with various deep learning algorithms to detect weapons from CCTV videos to provide security to civilians or tourist. All public places are equipped with CCTV and this CCTV can be monitor by using this deep learning algorithms to detect person with weapon. In propose paper author has various object detection and classification models called VGG16, Yolov3, Yolov4, faster RCNN, Inception, Resnet and few others. In all algorithms Yolov4 is giving more than 90% accuracy and FSCORE.

To train all those algorithms there is no public dataset available so author of this paper 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



So by using above images will trained algorithm

**MODULES:**

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

**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.

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

**6.2 SAMPLE CODE:**

|  |
| --- |
| # -\*- coding: utf-8 -\*- |
|  |  |
|  |  |
|  | #importing Keras, Library for deep learning |
|  | from keras.models import Sequential |
|  | from keras.layers.core import Dense, Dropout, Activation, Flatten |
|  | from keras.layers.convolutional import Convolution2D, MaxPooling2D |
|  | from keras.utils import np\_utils |
|  | from keras.preprocessing.image import img\_to\_array |
|  |  |
|  | import numpy as np |
|  |  |
|  | # Image manipulations and arranging data |
|  | import os |
|  | from PIL import Image |
|  | # import theano |
|  | # theano.config.optimizer="None" |
|  | #Sklearn to modify the data |
|  |  |
|  | from sklearn.model\_selection import train\_test\_split |
|  | # os.chdir("provide path") |
|  |  |
|  | # input image dimensions |
|  | m,n = 240,240 |
|  |  |
|  | path1='test\\' |
|  | path2='train\\' |
|  |  |
|  | classes=os.listdir(path2) |
|  | x=[] |
|  | y=[] |
|  | count = 0 |
|  | for fol in classes: |
|  | print (fol) |
|  | imgfiles=os.listdir(path2 + '\\' + fol); |
|  | for img in imgfiles: |
|  | try: |
|  | im=Image.open(path2+'\\'+fol+'\\'+img); |
|  | im=im.convert(mode='RGB') |
|  | imrs=im.resize((m,n)) |
|  | imrs=img\_to\_array(imrs)/255; |
|  | imrs=imrs.transpose(2,0,1); |
|  | imrs=imrs.reshape(3,m,n); |
|  | x.append(imrs) |
|  | y.append(count) |
|  | except: |
|  | pass |
|  | count += 1 |
|  |  |
|  | x=np.array(x); |
|  | y=np.array(y); |
|  |  |
|  | batch\_size=32 |
|  | nb\_classes=len(classes) |
|  | nb\_epoch=20 |
|  | nb\_filters=128 |
|  | nb\_pool=2 |
|  | nb\_conv=3 |
|  |  |
|  | x\_train, x\_test, y\_train, y\_test= train\_test\_split(x,y,test\_size=0.2,random\_state=4) |
|  |  |
|  | uniques, id\_train=np.unique(y\_train,return\_inverse=True) |
|  | Y\_train=np\_utils.to\_categorical(id\_train,nb\_classes) |
|  | uniques, id\_test=np.unique(y\_test,return\_inverse=True) |
|  | Y\_test=np\_utils.to\_categorical(id\_test,nb\_classes) |
|  |  |
|  |  |
|  | model= Sequential() |
|  | model.add(Convolution2D(nb\_filters,nb\_conv,nb\_conv,border\_mode='same',input\_shape=x\_train.shape[1:])) |
|  | model.add(Activation('relu')) |
|  | model.add(Convolution2D(int(nb\_filters/2),nb\_conv,nb\_conv,border\_mode='same')); |
|  | model.add(Activation('relu')) |
|  | #model.add(MaxPooling2D(pool\_size=(nb\_pool,nb\_pool))); |
|  | model.add(Dropout(0.2)) |
|  | model.add(Convolution2D(int(nb\_filters/4),nb\_conv,nb\_conv,border\_mode='same')); |
|  | model.add(Activation('relu')) |
|  | model.add(Convolution2D(int(nb\_filters/8),nb\_conv,nb\_conv,border\_mode='same')); |
|  | model.add(Activation('relu')) |
|  | model.add(MaxPooling2D(pool\_size=(nb\_pool,nb\_pool))); |
|  | model.add(Dropout(0.2)); |
|  | model.add(Flatten()); |
|  | model.add(Dense(128)); |
|  | model.add(Dropout(0.2)); |
|  | model.add(Dense(nb\_classes)); |
|  | model.add(Activation('softmax')); |
|  | model.compile(loss='categorical\_crossentropy',optimizer='sgd',metrics=['accuracy']) |
|  |  |
|  |  |
|  | nb\_epoch=60 |
|  | batch\_size=32 |
|  | model.fit(x\_train,Y\_train,batch\_size=batch\_size,nb\_epoch=nb\_epoch,verbose=1,validation\_data=(x\_test, Y\_test)) |
|  |  |
|  | model.save("model\_latest.h5",overwrite=True) |
|  |  |
|  | files=os.listdir(path1); |
|  |  |
|  | img=files[0] |
|  | print (img) |
|  | im = Image.open(path1 + img); |
|  | imrs = im.resize((m,n)) |
|  | imrs=img\_to\_array(imrs)/255; |
|  | imrs=imrs.transpose(2,0,1); |
|  | imrs=imrs.reshape(3,m,n); |
|  |  |
|  | x=[] |
|  | x.append(imrs) |
|  | x=np.array(x); |
|  | predictions = model.predict(x) |
|  | print (predictions) |
|  | print (model.summary()) |

**SOFTWARE ENVIRONMENT**

**7.SOFTWARE ENVIRONMENT**

# What is Python :-

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

* + [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)
  + GUI Applications (like Kivy, Tkinter, PyQt etc. )
  + Web frameworks like Django (used by YouTube, Instagram, Dropbox)
  + Image processing (like Opencv, Pillow)
  + Web scraping (like Scrapy, BeautifulSoup, Selenium)
  + Test frameworks
  + Multimedia

### Advantages of Python :-

Let’s see how Python dominates over other languages.

#### 1. Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

#### 2. Extensible

As we have seen earlier, Python can be**extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

#### 3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities**to our code in the other language.

#### 4. Improved Productivity

The language’s simplicity and extensive libraries render programmers**more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

#### 5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

#### 6. Simple and Easy

When working with Java, you may have to create a class to print **‘Hello World’**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and**code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

#### 7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

#### 8. Object-Oriented

This language supports both the **procedural and object-oriented**programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

#### 9. Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you[**download Python**](https://data-flair.training/blogs/install-python-windows/) for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

#### 10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to**code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

#### 11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

### **Advantages of Python Over Other Languages**

#### 1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

#### 2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

**The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.**

#### 3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and [**machine learning**](https://data-flair.training/blogs/machine-learning-tutorials-home/), automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

### **Disadvantages of Python**

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

#### 1. Speed Limitations

We have seen that Python code is executed line by line. But since [Python](https://www.python.org/) is interpreted, it often results in **slow execution**. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

#### 2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Brython is that it isn’t that secure.

#### 3. Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don’t need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can**raise run-time errors**.

#### 4. Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

#### 5. Simple

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

**History of Python : -**

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

**Python Development Steps : -**

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.  
Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode.Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

* Print is now a function
* Views and iterators instead of lists
* The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
* There is only one integer type left, i.e. int. long is int as well.
* The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
* Text Vs. Data Instead Of Unicode Vs. 8-bit

**Purpose :-**

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

**Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Modules Used in Project :-**

**Tensorflow**

TensorFlow is a [free](https://en.wikipedia.org/wiki/Free_software) and [open-source](https://en.wikipedia.org/wiki/Open-source_software) [software library for dataflow and differentiable programming](https://en.wikipedia.org/wiki/Library_(computing)) across a range of tasks. It is a symbolic math library, and is also used for [machine learning](https://en.wikipedia.org/wiki/Machine_learning) applications such as [neural networks](https://en.wikipedia.org/wiki/Neural_networks). It is used for both research and production at [Google](https://en.wikipedia.org/wiki/Google).‍

TensorFlow was developed by the [Google Brain](https://en.wikipedia.org/wiki/Google_Brain) team for internal Google use. It was released under the [Apache 2.0](https://en.wikipedia.org/wiki/Apache_License) [open-source license](https://en.wikipedia.org/wiki/Open-source_license) on November 9, 2015.

**Numpy**

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

**Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](http://ipython.org/) shells, the [Jupyter](http://jupyter.org/) Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the [sample plots](https://matplotlib.org/tutorials/introductory/sample_plots.html) and [thumbnail gallery](https://matplotlib.org/gallery/index.html).

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

**Scikit – learn**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. **Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Install Python Step-by-Step in Windows and Mac :**

Python a versatile programming language doesn’t come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

## How to Install Python on Windows and Mac :

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

**Note:** The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. [Download the Python Cheatsheet here.](https://myelearninghub.com/python-cheat-sheet/)The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

### Download the Correct version into the system

**Step 1:** Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: [**https://www.python.org**](https://www.python.org/)



Now, check for the latest and the correct version for your operating system.

**Step 2:** Click on the Download Tab.

****

**Step 3:** You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

****

**Step 4:** Scroll down the page until you find the Files option.

**Step 5:** Here you see a different version of python along with the operating system.



• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

**Note:** To know the changes or updates that are made in the version you can click on the Release Note Option.

### Installation of Python

**Step 1:** Go to Download and Open the downloaded python version to carry out the installation process.



**Step 2:** Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



**Step 3:** Click on Install NOW After the installation is successful. Click on Close.



With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

**Note:** The installation process might take a couple of minutes.

### Verify the Python Installation

**Step 1:** Click on Start

**Step 2:** In the Windows Run Command, type “cmd”.



**Step 3:** Open the Command prompt option.

**Step 4:** Let us test whether the python is correctly installed. Type **python –V** and press Enter.



**Step 5:** You will get the answer as 3.7.4

**Note:** If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

### Check how the Python IDLE works

**Step 1:** Click on Start

**Step 2:** In the Windows Run command, type “python idle”.



**Step 3:** Click on IDLE (Python 3.7 64-bit) and launch the program

**Step 4:** To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



**Step 5:** Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

**Step 6:** Now for e.g. **enter print**.

**SYSTEM TESTING**

**8.SYSTEM TESTING**

**8.1 TESTING STRATEGIES**

**UNIT TESTING**

Unit testing, a testing technique using which individual modules are tested to determine if there are issues by the developer himself.. it is concerned with functional correctness of the standalone modules. The main aim is to isolate each unit of the system to identify, analyze and fix the defects.

Unit Testing Techniques:

Black Box Testing - Using which the user interface, input and output are tested.

White Box Testing –Used to test each one of those functions behavior is tested.

**DATA FLOW TESTING**

Data flow testing is a family of testing strategies based on selecting paths through the program’s control flow in order to explore sequence of events related to the status of Variables or data object. Dataflow Testing focuses on the points at which variables receive and the points at which these values are used.

**INTEGRATION TESTING**

Integration Testing done upon completion of unit testing, the units or modules are to be integrated which gives raise too integration testing. The purpose of integration testing is to verify the functional, performance, and reliability between the modules that are integrated.

**BIG BANG INTEGRATION TESTING**

Big Bang Integration Testing is an integration testing Strategy wherein all units are linked at once, resulting in a complete system. When this type of testing strategy is adopted, it is difficult to isolate any errors found, because attention is not paid to verifying the interfaces across individual units.

**USER INTERFACE TESTING**

User interface testing, a testing technique used to identify the presence of defects is a product/software under test by Graphical User interface [GUI].

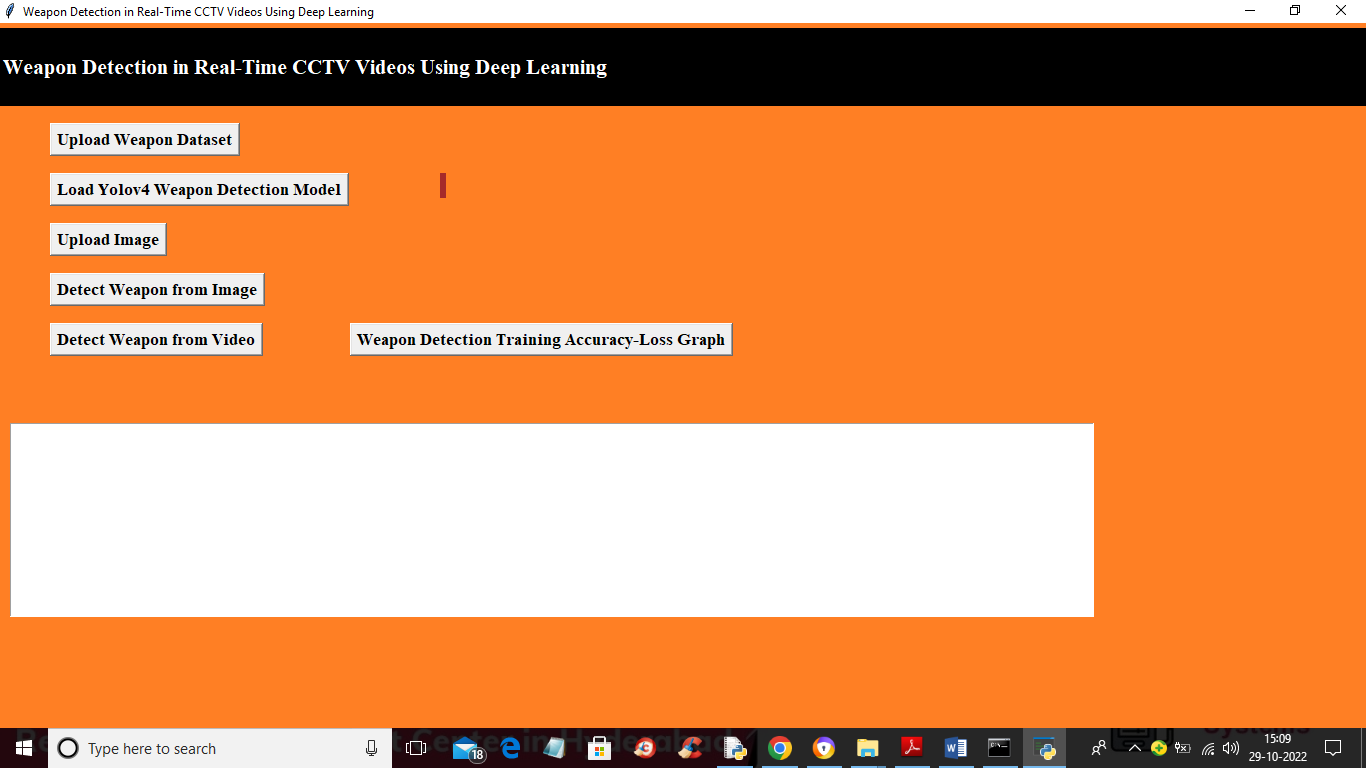
**8.2 TEST CASES:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **INPUT** | **If available** | **If not available** |
| 1 | Upload Weapon Dataset | Dataset loaded | There is no process |
| 2 | Load Yolov4 Weapon Detection Model | load Yolov4 model and then calculate its prediction accuracy | There is no process |
| 3 | Upload Image | will upload test image | There is no process |
| 4 | Detect Weapon from Image | apply Yolov4 model on loaded image to detect weapon | There is no process |
| 5 | Detect Weapon from Video | upload video and then YoloV4 will analyse each frame in the video to detect weapon | There is no process |
| 6 | Weapon Detection Training Accuracy-Loss Graph | plot YoloV4 training accuracy and loss graph | There is no process |

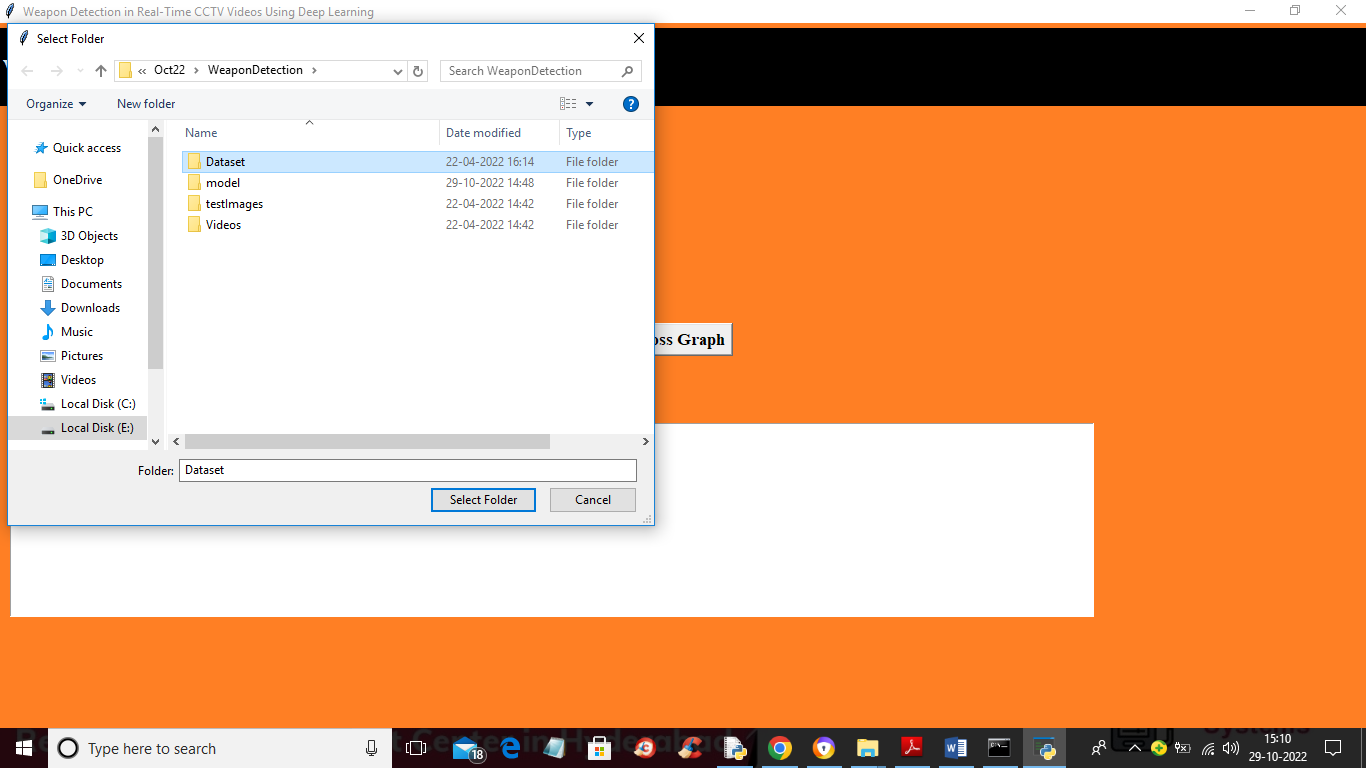
**SCREENS**

1. **SCREENSHOTS**

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



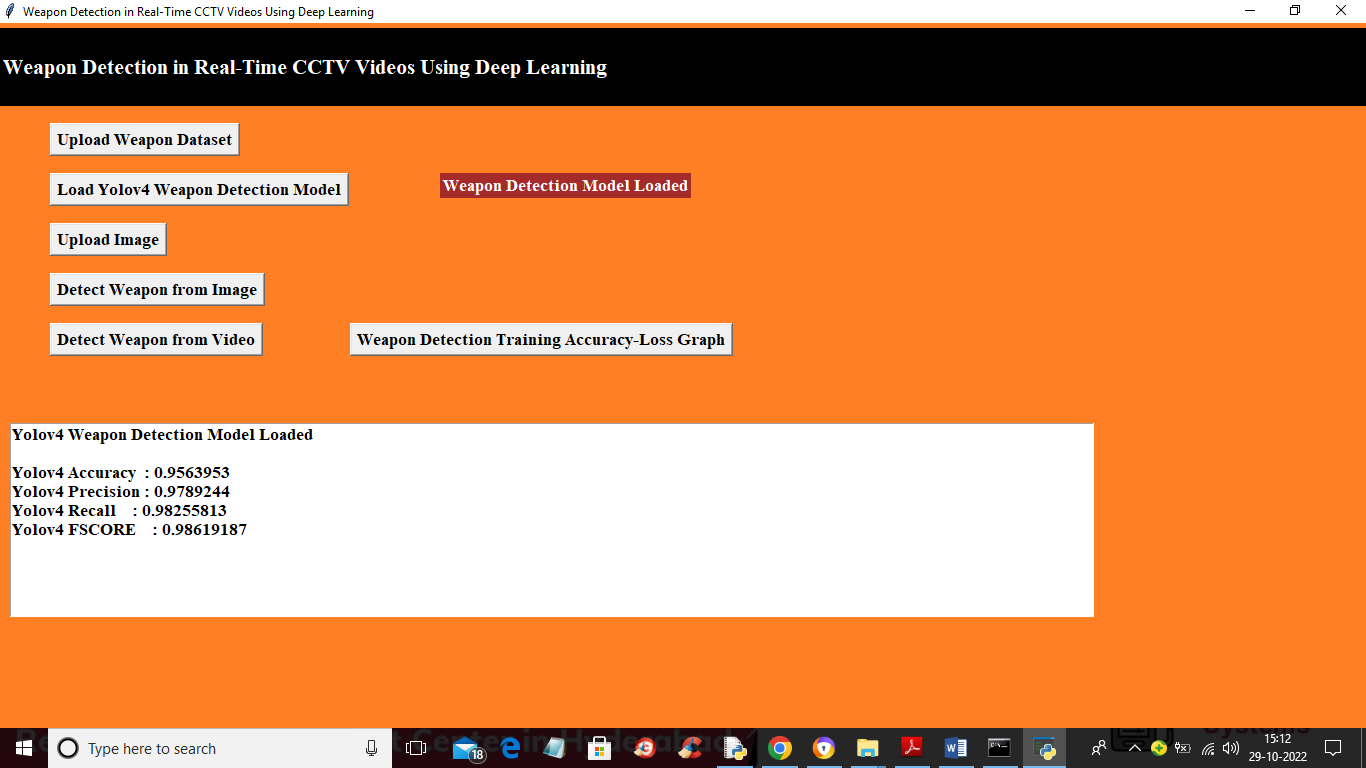
In above screen click on ‘Upload Weapon Dataset’ button to upload dataset and get below output



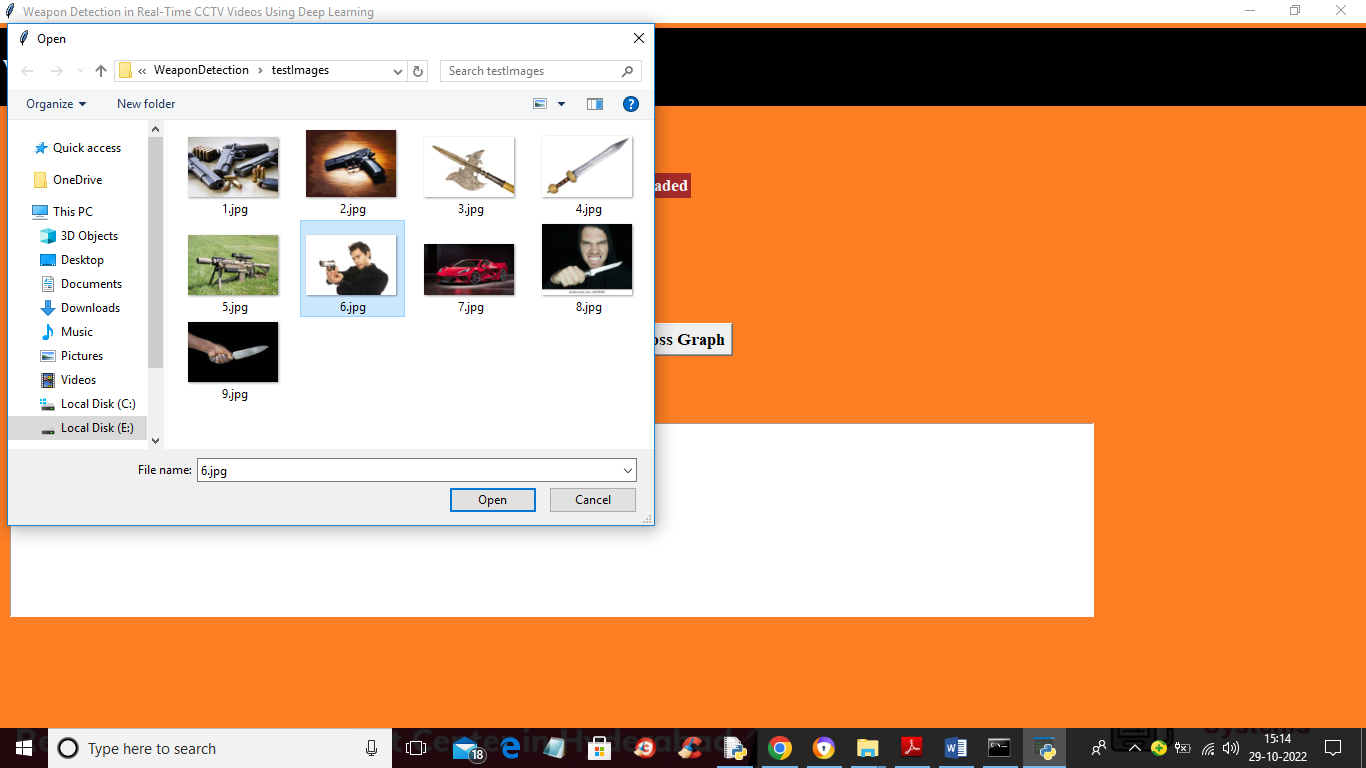
In above screen selecting and uploading ‘Dataset’ folder and then click on ‘Select Folder’ button to load dataset and get below output



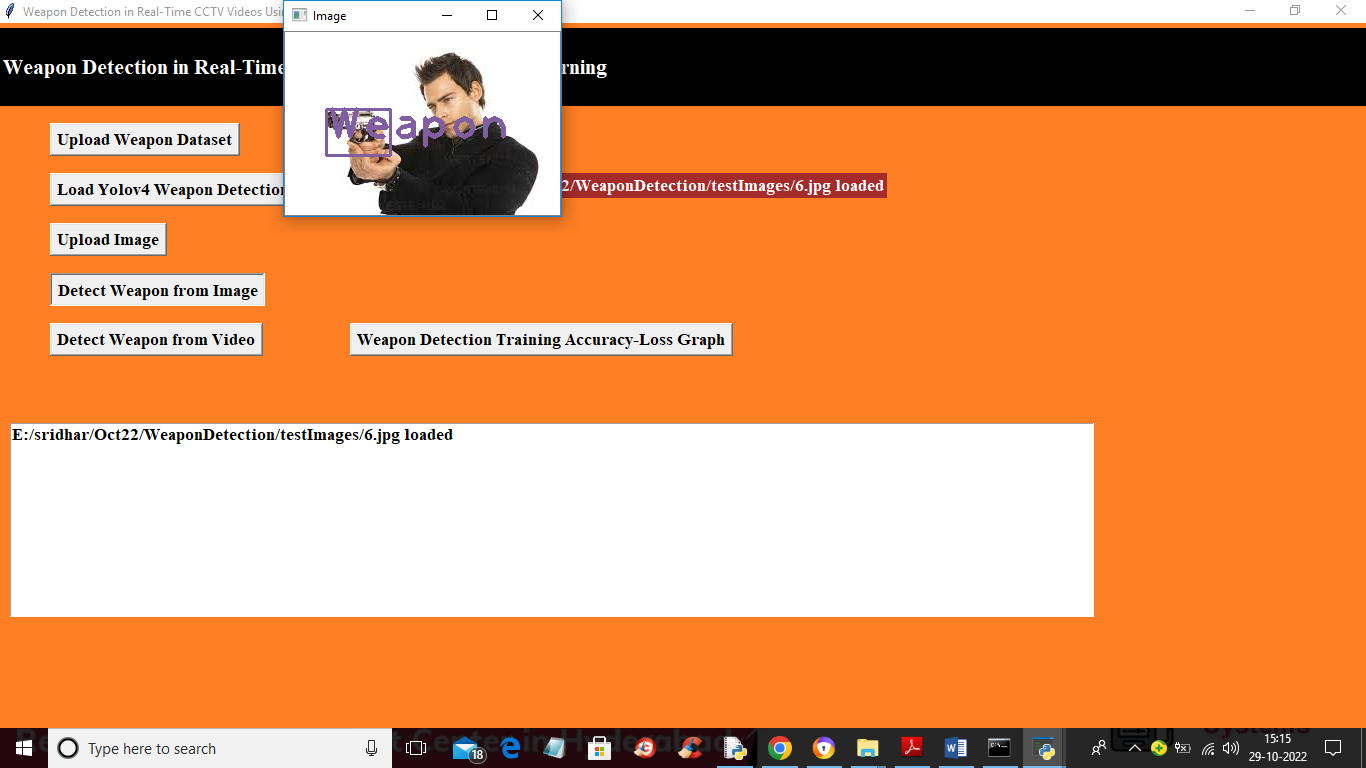
In above screen we can see dataset loaded and dataset contains 2 classes called Guns and Knives and dataset contains total 3635 images and now close above image and then click on ‘Load Yolov4 Weapon Detection Model’ button to load YoloV4 model and calculate accuracy



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

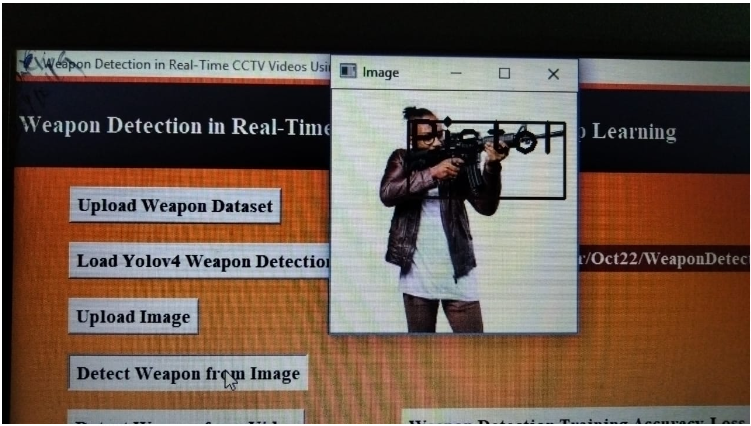


In above screen selecting and uploading ‘6.jpg’ image and then click on ‘Open’ button to load image and then click on ‘Detect Weapon from Image’ button to detect weapon and get below output

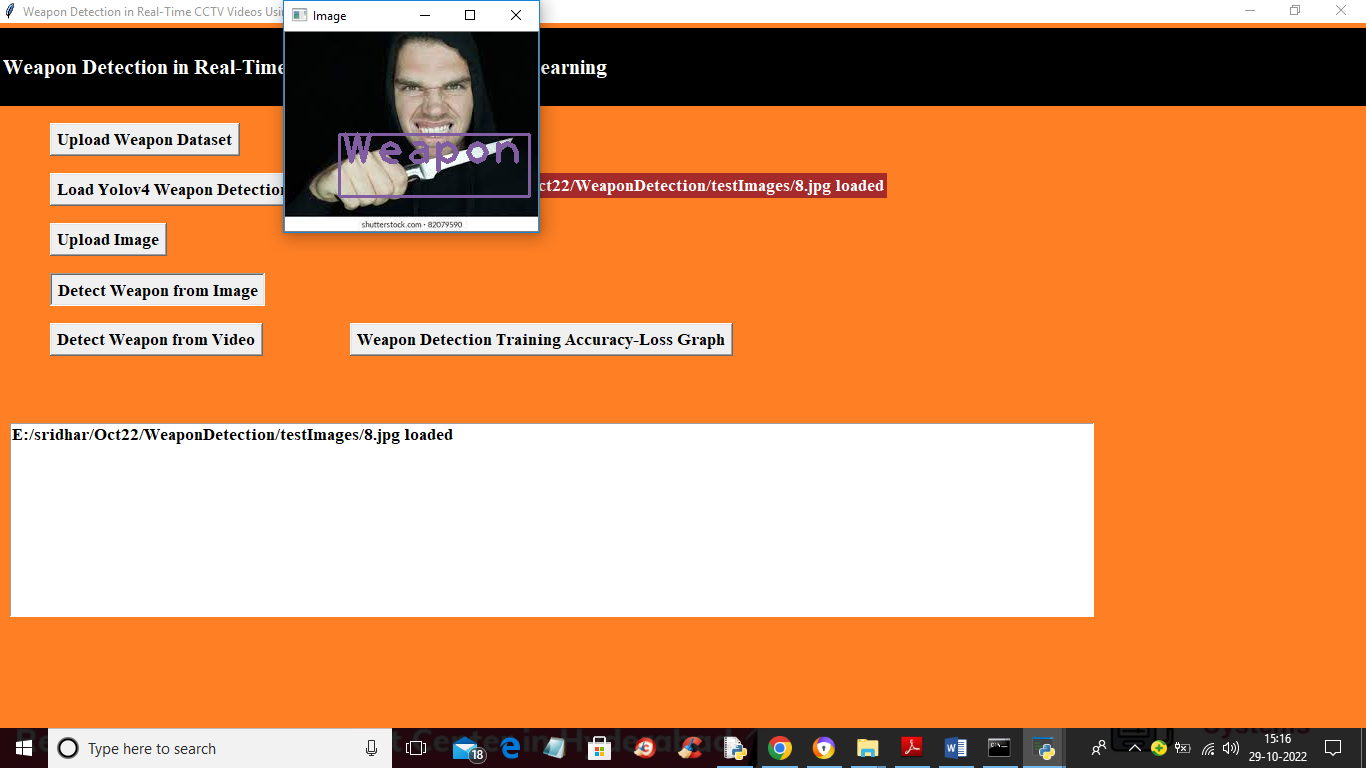


In above screen we got bounding box with label as ‘Weapon’ detect and will get beep sound also and similarly you can upload any image and get detection

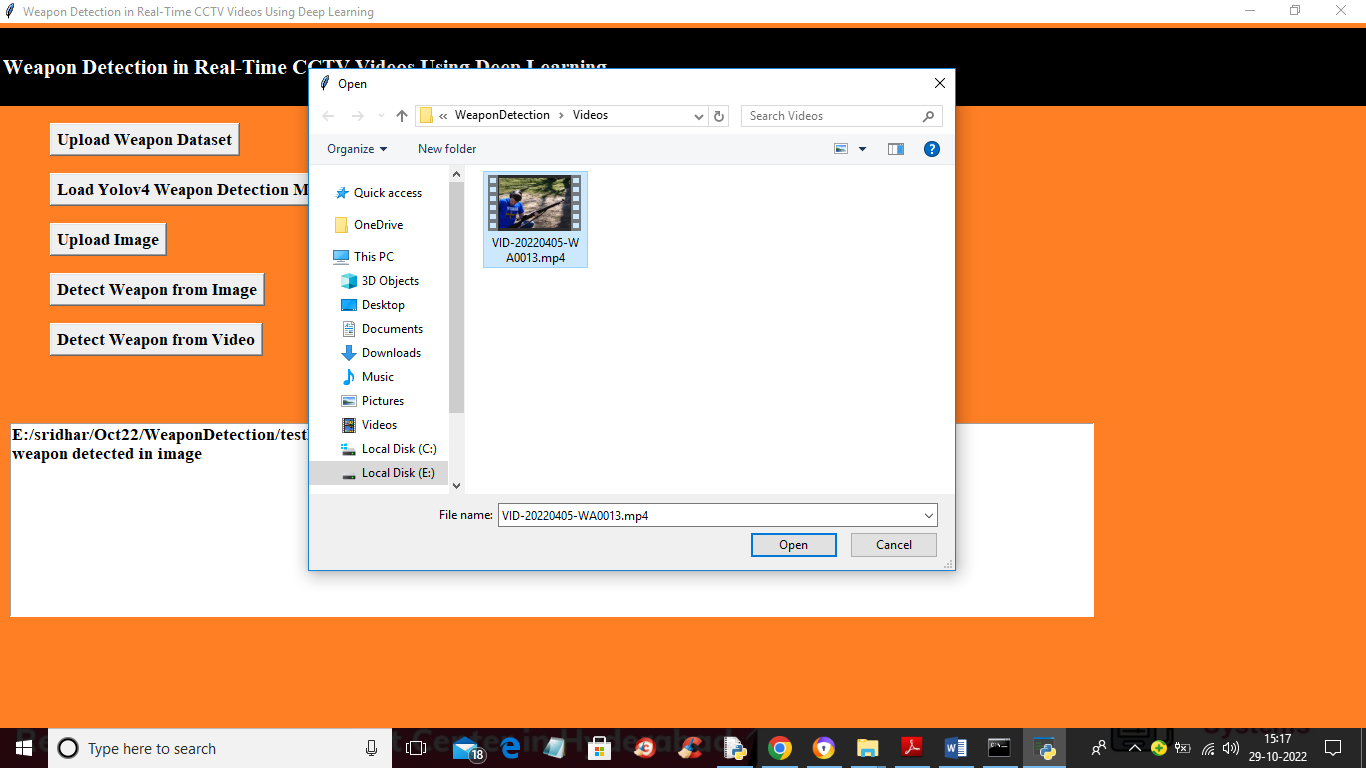




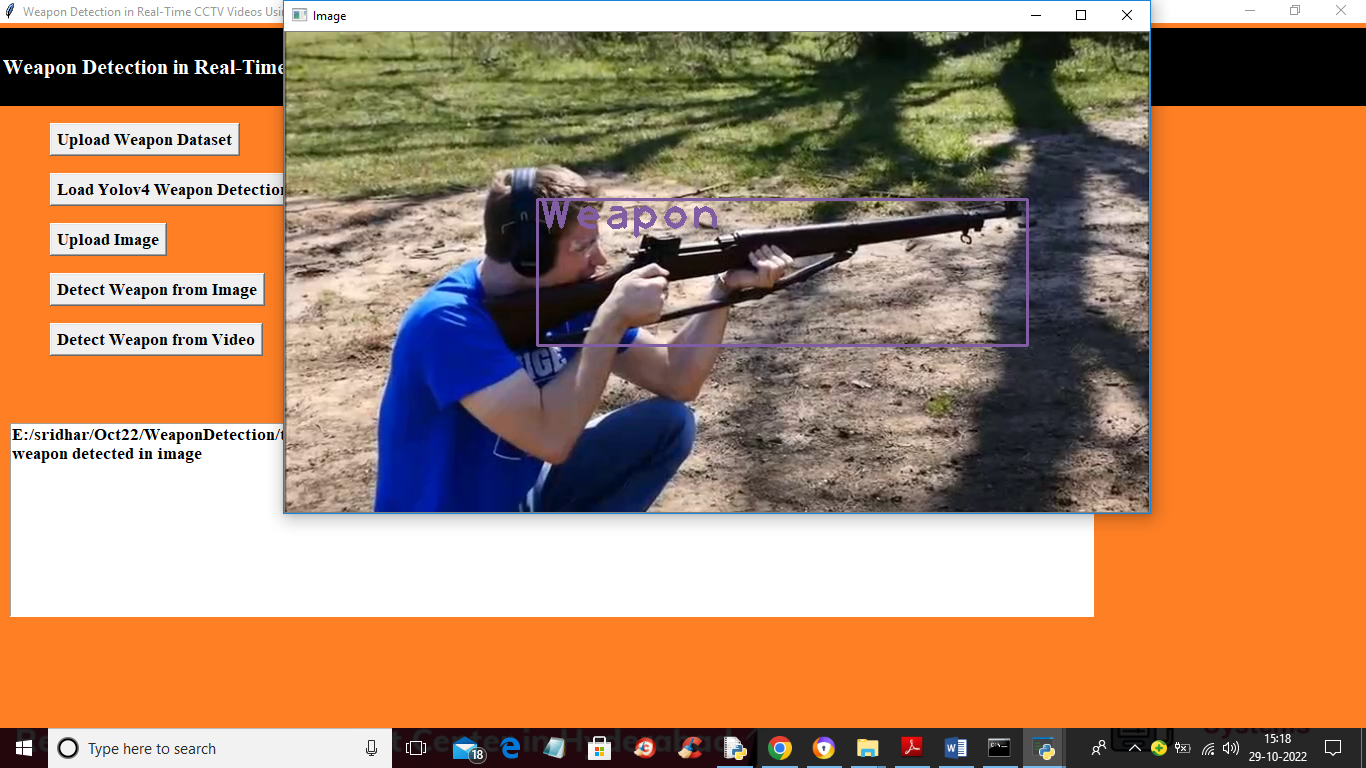




In above image also weapon detected and now close above image and then click on ‘Detect Weapon from Video’ button to upload video and detect weapon from it



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



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



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

**CONCLUSION**

**10.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.

**BIBILOGRAPHY**

**11. REFERENCES**

[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. Matiola“ski, 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.