A PROJECT REPORT ON

“ Effective Usage Of Artificial Intelligence And Deep Learning For Implementing Weapon Detection For Security Applications”

A Project report submitted in partial fulfillment of the requirement for the award ofdegree of

BACHELOR OF TECHNOLOGY

IN

CSE (ARTIFICIAL INTELLIGENCE)

By

SHAIK RUHEENA TANVEER	**20AM1A3127**
VIGNESWAR REDDY MEKALA	**20AM1A3155**
SHASHIKALA VANIPENTA	**20AM1A3137**

Under the Esteemed guidance of

Mr.T.NIRANJAN BABU **M.Tech,(Ph.D)**

DEPARTMENT OF CSE(ARTIFICIAL INTELLIGENCE)
SVR ENGINEERING COLLEGE
AYYALURU METTA, NANDYAL, KURNOOL(DIST.)-518503
(Approved by AICTE, New Delhi & Affiliated to JNTU,
Anantapuram) are accredited by NBA.
ACADEMIC YEAR: 2020-2024

SVR ENGINEERING COLLEGE

AYYALURU METTA, NANDYAL, KURNOOL(DIST.)-518503
(Approved by AICTE, New Delhi & Affiliated to JNTU,
Anantapuram)accredited by NBA.**ACADEMIC
YEAR: 2020-2024**

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE)

CERTIFICATE

This is certified that the technical seminar entitled, **"Effective Usage Of Artificial Intelligence And Deep Learning For Implementing Weapon Detection For Security Applications"** is being submitted by **S.RUHEENA TANVEER(20 AM1A3127), VIGNESWAR REDDY MEKALA (20AM1A3155),SHASHIKALA VANIPENTA (20AM1A3137)** in partial fulfillment of the requirements for the award of Degree of BACHELOR OF TECHNOLOGY in CSE(ARTIFICIAL INTELLIGENCE),SVR ENGINEERING COLLEGE is a record of bonafide work carried out by them during the academic year 2020-2024 under our guidance and supervision. The results presented in thistechnical seminar have been verified and found to be satisfactory.

Signature of the Guide	Head of the Department
Mr.T.NIRANJAN BABU M.Tech,(Ph.D) Assistant Professor Department of CSE(AI)	**Mr. M.SUBBA REDDY, M.Tech.,(Ph.D)** Associate Professor Department of CSE(AI)

External Examiner

STUDENT DECLARATION

We here by declare that the project report entitled “ **EFFECTIVE USAGE OF ARTIFICIAL INTELLIGENCE AND DEEP LEARNING FOR IMPLEMENTING WEAPON DETECTION FOR SECURITY APPLICATIONS** ” is carried out by us during the academic year 2023-2024 in partial fulfilment of the award of Bachelor of Technology in CSE (ARTIFICIAL INTELLIGENCE) from SVR Engineering College affiliated to Jawaharlal Nehru Technological University Anantapur. We have no submitted the same to any other university or organization for the award of any other degree.

By

SHAIK RUHEENA TANVEER (20AM1A3127)

VIGNESWAR REDDY MEKALA (20AM1A3155)

SHASHIKALA VANIPENTA (20AM1A3137)

ACKNOWLEDGEMENT

We earnestly take the responsibility to acknowledge the following distinguished personalities who graciously allowed us to carry out project work successfully.

We express deep gratitude to our guide **Mr.T.NIRANJAN BABU M.Tech,(Ph.D),** Assistant Professor, Department of CSE(AI), **SVR Engineering College**, for the guidance and encouragement throughout the course of the project work. His friendly and informal talks helped us to work under excellent working conditions.

We would like to express our gratitude to project coordinator **Mrs. B.V.S.NAGA LAKSHMI, M.Tech ,** Assistant Professor in the Department of CSE(AI), **SVR EngineeringCollege,** for their encouragement throughout the course.

We would like to express our gratitude to senior faculty **Mr. B.RAMA SUBBAIAH, M.Tech ,(Ph.D),** Associate Professor in the Department of CSE(AI), **SVR Engineering College,** for their encouragement throughout the course.

We are thankful to the Head of the Department of CSE(AI), **Mr. M.SUBBA REDDY, M.Tech,(Ph.D),** Associate Professor **SVR Engineering College**, for the encouragement and assistance provided to us, which contributed to the successful completion of this project.

We are thankful to our Principal **Dr.P. MALLIKARJUNA REDDY** who has encouraged and motivated us to complete the project by providing all the necessary facilitiesto carry out the work in the college.

We are thankful to our Honourable Chairman **Sri S.V. RAMI REDDY** & Honourable Managing Director **Sri S. DINESH REDDY** for providing good faculty and fortheir moral support throughout the course.

We would like to thank all teaching and non-teaching members of the CSE(AI) Department for their generous help in various ways for the completion of this thesis.
They have been great sources of inspiration to us and we thank them from the bottom of myheart.

By

S.RUHEENA TANVEER(20AM1A3127)

VIGNESWAR REDDY MEKALA(20AM1A3155)

SHASHIKALA VANIPENTA(20AM1A3137)

SVR ENGINEERING COLLEGE

Service Virtue Reverence

Approved by AICTE, New Delhi, permanently affiliated to JNTUA, Anantapuramu

Accredited by NBA, Ayyaluru metta, Nandyal (Dist.), AP-518501

DEPARTMENT OF CSE (ATIFICIAL INTELLIGENCE)

INSTITUTE VISION, MISSION AND CORE VALUES

VISION

To produce Competent Engineering Graduates & Managers with a strong base of Technical & Managerial Knowledge and the Complementary Skills needed to be Successful Professional Engineers & Managers.

MISSION

To fulfill the vision by imparting Quality Technical & Management Education to the Aspiring Students, by creating Effective Teaching/Learning Environment and providing State – of the – Art Infrastructure and Resources.

CORE VALUES

- Commitment to Continual Improvement
- Inclusiveness
- Enhancement of Student Learning through Innovative Instructional Methods
- Collaboration with Industry and other Institutions for Mutual Benefits Pursuit of Excellence in all Activities.

SVR ENGINEERING COLLEGE

Service Virtue Reverence

Approved by AICTE, New Delhi, permanently affiliated to JNTUA, Ananthapuramu
Accredited by NBA, Ayyaluru metta, Nandyal(DT), AP-518502

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE)

DEPARTMENT VISION AND MISSION

VISION

To build an ecosystem to contribute to society by producing leaders in Artificial Intelligence through quality education.

MISSION

- Train the students in the state-of-the-art technologies of AI.
- Sensitize students to solve societal issues through AI techniques by inculcating values and ethics.
- Enhance employability and entrepreneurial skills in the field of AI through experiential and self-directed learning

PO Number	Graduate Attributes	PO Statements
1	Engineering Knowledge	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex Engineering problems.
2	Problem Analysis	Identify, formulate, review research literature, and analyse Complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3	Design/ Development of Solutions	Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety ,and the cultural, societal ,and environmental considerations.
4	Conduct Investigations of Complex Problems	Ability to review research literature, use research methods to execute project and synthesize the problem to provide valid conclusions.
5	Modern Tool Usage	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6	The Engineer and Society	Apply reasoning informed by the contextual Knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7	Environment and Sustainability	Understand the impact of the professional engineering solutions in societal and environmental contexts, and Demonstrate the knowledge of, and need for sustainable development.
8	Ethics	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9	Individual and team work	Function effectively as an individual, and as a member or leader in di verse teams, and in multidisciplinary settings.
10	Communication	Communicate effectively on complex engineering activities with the engineering community and with society at large , such as, being able to comprehend and write effective reports and design documentation ,make effective presentations ,and give and receive clear instructions.

11	Project Management and Finance	Demonstrate knowledge and understanding of the engineering and management principles and apply these to one 's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12	Life -long Learning	Recognize the need for, and have the preparation and ability to engage in Independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes

S.NO	PSO Number	Program Specific Outcomes
1	PSO 1	Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
2	PSO 2	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest contextof technological change.

PROJECT CO-PO MAPPING:

TITLE	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
EFFECTIVE USAGE OF AI & DL FOR IMPLEMENTING WEAPON DETECTION FOR SECURITY APPLICATIONS	**3**	**2**	**3**	**2**	**2**	**3**	**1**	**3**	**2**	**1**	**2**	**3**	**2**	**3**

SIGN OF GUIDE **SIGN OF H.O.D**

ABSTRACT

Because crime rates are rising in crowded places and suspiciously isolated locales, security remains a top priority in all fields. The principal uses of computer vision to address a variety of issues include abnormal detection and monitoring. The requirement for deployment for video surveillance systems that can identify and analyze the scene and anomalous events play a crucialrole for intelligence monitoring due to the growing demand in the safeguarding of safety,security,and personal property. Using Faster RCNN techniques, this research implementsautomatic detection of guns or other weapons. The two kinds of data used in the proposed solution. There was a pre-labeled image data set and a manually labeled set of photos in the other data set. The algorithms yield tabularresults with good accuracy; however,the trade-off in speed and precision may determine how these algorithms are applied in practical scenarios.

Keywords: RCNN technique, safeguarding, surveillance system

CONTENTS

CONTENT NAME	PAGE NO
1. INTRODUCTION	**01**
1.1 Objective of the project	
2. LITERATURE SURVEY	**02-07**
3. SYSTEM ANALYSIS	**08-19**
3.1 Existing system	
3.2 Proposed system	
3.3 Process model used with justification	
3.4 Software Requirement Specification	
3.4.1 Overall Description	
3.4.2 External Interface Requirement	
4. SYSTEM DESIGN	**20-29**
5. IMPLEMENTATION	**30-47**
5.1 Python	
5.2 Sample Code	
6. SCREENSHOTS	**49-55**
7. CONCLUSION	**56-57**
8. REFERENCE	**58-59**

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
3.3.1	SDLC Architecture	10
3.3.2	Requirement Stages in SDLC	11
3.3.3	Analysis Stages in SDLC	12
3.3.4	Designing Stages in SDLC	13
3.3.5	Development Stages in SDLC	14
3.3.6	Integration & Test Stages in SDLC	15
3.3.7	Installation & Acceptance Test in SDLC	16
4.1	Class Diagram	22
4.2	Sequence Diagram	23
4.3	Collaboration Diagram	24
4.4	Component Diagram	25
4.5	Deployment Diagram	26
4.6	Activity Diagram	27
4.7	Data Flow Diagram	28

SCREENSHOT NO	SCREENSHOT NAME	PAGE NO
6.1	Uploading the Dataset	50
6.2	Selecting The Folder	50
6.3	Generate & Load Weapon Detection Model	51
6.4	Uploading Images	51
6.5	Detecting Weapon From Images	52
6.6	Detecting Weapon From Videos	52
6.7	Uploading MP4Video Files	53
6.8	Detecting The Weapon	53
6.9	FRCNN Weapon Detection Training Accuracy Loss Graph	54

CHAPTER-1

INTRODUCTION

1.INTRODUCTION

Weapon or Anamoly detection is the identification of irregular, unexpected unpredictable,unusual events or items, which is not considered as a normally occurring event or a regular itemin a pattern or items present in a dataset and thus different from existing patterns. An anomaly isa pattern that occurs differently froma set of standard patterns. Therefore,anomalies depend on the phenomenon of interest. Object detection uses feature extraction andlearning algorithms or models to recognize instances of various category of objects. Proposed implementation focuses on accurate gun detection and classification. Also concerned with accuracy, since a false alarm could result in adverse responses. Choosing the right approach required to make a proper trade- off between accuracy and speed. Frames areextracted from the input video. Frame differencing algorithm is applied and bounding box created before the detection of object.

Dataset is created, trained and fed to object detection algorithm. Based on application suitable detection algorithm (SSD or fast RCNN) chosen for gun detection. The approach addresses a problem of detection using various machine learning models like Region Convolutional Neural Network (RCNN), Single Shot Detection (SSD).

1.1 Objective of the project:

Security is always a main concern in every domain, due to a rise in crime rate in a crowded event or suspicious lonely areas. Abnormal detection and monitoring havemajor applications of computer vision to tackle various problems. Due to growing demand in the protection of safety, security and personal properties, needs and deployment of video surveillance systems can recognize and interpret the scene andanomaly events play a vital role in intelligence monitoring. This paper implements automatic gun (or) weapon detection using a convolution neural network (CNN) based SSD and Faster RCNN algorithms. Proposed implementation uses two types of datasets. One dataset, which had pre-labelled images and the other one is a set ofimages, which were labelled manually. Results are tabulated, both algorithms achieve good accuracy, but their application in real situations can be based on the trade-off between speed and accuracy.

CHAPTER-2

LITERATURE SURVEY

1. LITERATURE SURVEY

“Scalable Object Detection Using Deep Neural Networks”:

Deep convolutional neural networks have recently achieved state-of-the-art performance on a number of image recognition benchmarks, including the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC-2012). The winning model on the localization sub-task was a network that predicts a single bounding box and a confidence score for each object category in the image. Such a model captures the whole-image context around the objects but cannot handle multiple instances of the same object in the image without naively replicating the number of outputs for each instance. In this work, we propose a saliency- inspired neural network model for detection, which predicts a set of class-agnostic bounding boxes along with a single score for each box, corresponding to its likelihood of containing any object of interest. The model naturally handles a variable number of instances for each class and allows for crossclass generalization at the highest levels of the network.

We are able to obtain competitive recognition performance on VOC2007 andILSVRC2012, while using only the top few predicted locations in each image and a small number of neural network evaluations **“Anomaly Detection in Videos for Video Surveillance Applications Using Neural Networks,”**

Security is always a main concern in every domain, due to a rise in crime rate in the crowded eventor suspicious lonely areas. Abnormal detection and monitoring have major applications of computer vision to tackle various problems. Due to growing demand in the protection of safety, security and personal properties, the needs and deployment of video surveillance systems can recognize and interpret the scene andanomaly events play a vital role in intelligence monitoring. Anomaly detection is a technique used to distinguish various patterns and identify unusual patterns with a minimal period, this pattern is called outliers. Surveillance videos can capture a variety of realistic anomalies. Anomaly detection in video surveillance involves breaking down the whole process into three layers, which are video labelers, image processing, and activity detection. Hence, anomaly detection in videos for video surveillance application gives assured results in regards to real-time scenarios. In this paper, we anomaly was detected in images and videos with an accuracy of 98.5%

“A Review of Artificial Intelligence Methods for Data Science and Data Analytics: Applications and Research Challenges,”

Artificial intelligence is a field which requires multidisciplinary expertise where thefinal goal is to automate all the human activities that presently require human intelligence. The majorproblem is to develop a method which works exactly the way how a human brain works. The architecture of artificial intelligence must emphasize on evaluation and redesign the nature of design process. Data science is also trending now and analytically deals to solve complex problems. Data is divided into smaller parts and its trends, behaviors are understood. The main problem in data science is to handle large quantities of data. Though there is significant increase in terms of research opportunities few challenges like lack of compute power, people power stillremains a big challenge.

“Classification of Objects in Video Records using Neural Network Framework,” Object Classification is a principle task in image and video processing. It is exercised over a multitude of applications ranging from test and numberclassification to traffic surveillance. The primitive machine learning concepts had provided the pedestal for carrying out umber of image processing tasks. Classifier such as Haar cascade which uses Haar like features was primitively used for face detection. Nowadays it's used for tacking and detection purposes all image processing tasks. Classifier such as Haar cascade which uses Haar like features was primitively used for face detection. Nowadays it's used for tacking and detection purposes also.Moreover, due to the ever-increasing demand and scope of improvement in the existing fields, the primitive methods need a lot of upgradation. Neural Networks have made the tasks quite plain sailing. Right from the vanilla neural networks to Fast R-CNN and then Faster R-CNN, all models have contributed significantly in the domain of computer vision. This paper mainly focuses in detection and classification ranging from single class objects to multi class objects. The Haar cascade classifier was trained on a batch of positive and negative samples which were later stitched together to form a vector file and finally form the xml file. On the other hand,

COCO dataset used for implementing R-CNN algorithm due to the presence of pertained model in it.

“Simulation and Performance Analysis of Feature Extraction and MatchingAlgorithms for Image Processing Applications”

Feature extraction and matching has the limelight in all almost all the fields ranging from biomedical to exploratory research. It has ubiquitous applications in present world that is moving at a breathtaking pace towards automation. The algorithms used for feature extraction are application specific, i.e. the one that yieldsbetter performance for face recognition does not guarantee the same performance for lane detection. A lot of time is invested in identifying algorithms that are best suited for an application. In the interest of time, an attempt has been made to develop few good algorithm combinations that assist in the selection of algorithms.The features of input image and the target image are extracted, described and matched using various algorithm combinations like SURF, FAST, MSER, and Harris Corner Detector. The combination of all these algorithms is simulated on MATLAB using computer vision and image processing toolboxes. A Graphical User Interface (GUI) is developed for better user experience. Face recognition is considered as an example to perform the simulation. The results reflect that the combination of SURF and MSER performs better compared to other algorithm combinations when an image is scaled and rotated, however there are no good matches when there is also a pose variation. Proper thresholding of ‘Match Threshold’, ‘Reject Ratio’, and ‘Inlier Threshold’ must be carried out through trial and error method to get better results. **“Simulation of Object Detection Algorithms for Video Survillance Applications”**

The plan is to establish an integrated system that can manage high-quality visual information and also detect weapons quickly and efficiently. It is obtained by integrating ARM-based computer vision and optimization algorithms with deep neural networks able to detect the presence of a threat. The whole system is connected to a Raspberry Pi module, which will capture live broadcasting and evaluate it using a deep convolutional neural network. Due to the intimate interaction between object identification and video and image analysis in real- timeobjects, By generating sophisticated ensembles that incorporate various low- level picture features with high-level information from object detection and scenario classifiers, their performance can quickly plateau. Deep learning models, which can learn semantic, high-level, deeper features, have been developed to overcome the issues that are present in optimization algorithms. It presents a review of deep learning based object detection frameworks that use Convolutional Neural Network layers for better understanding of object detection. The Mobile- Net SSD model behaves differently in network design, training methods, and optimization functions, among other things. The crime rate in suspicious areas has been reduced as aconsequence of weapon detection. However, security is always a major concernin human life.

The Raspberry Pi module, or computer vision, has been extensively used in the detection and monitoring of weapons. Due to the growing rate of human safety protection, privacy and the integration of live broadcasting systems which can detect and analyse images, suspicious areas are becoming indispensable in intelligence. This process uses a Mobile-Net SSD algorithm to achieve automatic weapons and object detection.

“Background Modelling techniques for foreground detection and Tracking usingGaussian Mixture model”

Background Modelling and Foreground detection in sports has been achieved by cleverly developing a model of a background from a video by deducing knowledge from frames and comparing this model to every subsequent frame and subtracting the background region from it, hence leaving the foreground detected. This output from GMM background subtraction is fed into the feature extraction algorithm, which segregates the players based on teams. By extracting information of primary colors from each frame, the design of the algorithm based on the color of preferenceis done. Tracking algorithms Kalman and extended Kalman Filters helpto predict and correct the location of players and in correctly estimating their trajectory on the field. Challenges such as shadowing, occlusions and illumination changes are addressed. The designed algorithms are tested against a set of performance parameters for the following datasets (Norway and FIFA) using MATLAB (2017b)and the inferences are respectively made. Object detection, motion detection and Kalman filter algorithms are implemented and theobserved results are 100%, 84% and 100% accuracy respectively. With the results quantification and performance analysis, it is observed that with the decrease in contrast between player jerseys a decrease in detection accuracy occurs and with players crowded regions on thefield and occluded players a decrease in tracking accuracy was observed.

“A metric for distributions with applications to image databases”

We introduce a new distance between two distributions that we call the Earth Mover's Distance (EMD), which reflects the minimal amount of work that must be performed to transform one distribution into the other by moving "distribution mass" around. This is a special case of the transportation problem from linear optimization, for which efficient algorithms are available. The

EMD also allows for partial matching. When used to compare distributions that have the same overall mass, the EMD is a true metric, and has easy-to-compute lower bounds. In this paper we focus on applications to image databases, especially color and texture. We use the EMD to exhibit the structure of color-distribution and texture spaces by means of Multi- Dimensional Scaling displays. We also propose a novel approachto the problem of navigating through a collection of color images, which leads to a new paradigm for image database search. **"Performance Analysis of Object Detection and Tracking Algorithms for Traffic Surveillance Applications using Neural Networks,"**

The single object detection has been performed by using the concepts of convolution layers. A neural network consists of several different layers such as the input layer, at least one hidden layer, and an output layer. The dataset used for single object detection is the on-road vehicle dataset. This dataset consists of three classes of images which are Heavy, Auto and Light. The dataset consists of images of varying illuminations. The performance metrics has been calculated for the day dataset, evening dataset and night dataset. Multiple object detection has been performed using the You Only Look Once (YOLOv3) algorithm. This approach encompasses a single deep convolution neural network dividing the input into a cell grid and each cell predicts a boundary box and classifies object directly. The dataset used for multiple object detection is the KITTI dataset. It consists of 80 classes out of which five classes has been considered for this project which are: car, bus, truck, and motorcycle and train. Using the Multiple Object Detection concepts, tracking of vehicles was further implemented. The first frame of the video was taken and Multiple object detection was performed and in the further frames of the video the object was tracked using its centroid position. This has been developed using OpenCV and Python using YOLOv3 algorithm for the object detection phase.

CHAPTER-3

SYSTEM REQUIREMENTS

2.SYSTEM ANALYSIS

1.1 EXISTING SYSTEM:

Abnormal and monitoring have major applications of computer vision to tacklevarious problems. Due to growing demand in the protection of safety, security andpersonalproperties, needs and deployment of video surveillance systems can recognize and interpret thescene and anomaly events play a vital role in intelligence monitoring.

This paper implementsautomatic gun (or) weapon detection using a convolution neuralnetwork (CNN)

DISADVANTAGES:

- Fast is lesser than FRCNN.

1.2 PROPOSED SYSTEM:

This paper implements automatic gun (or) weapon detection using based SSD and Faster RCNN algorithms. Proposed implementation uses two types of datasets. One dataset, which had pre-labelled images and the other one is a set of images, which were labelled manually. Results are tabulated, both algorithms achieve good accuracy, but their application in real situations can be based on the trade-off between speed and accuracy

ADVANTAGES:

- More Accuracy.
- FRCNN is faster than CNN

3.3. PROCESS MODEL USED WITH JUSTIFICATIONSDLC (Umbrella Model):

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

Stages in SDLC:

♦ Requirement Gathering

♦ Analysis

- ♦ Designing
- ♦ Coding
- ♦ Testing
- ♦ Maintenance

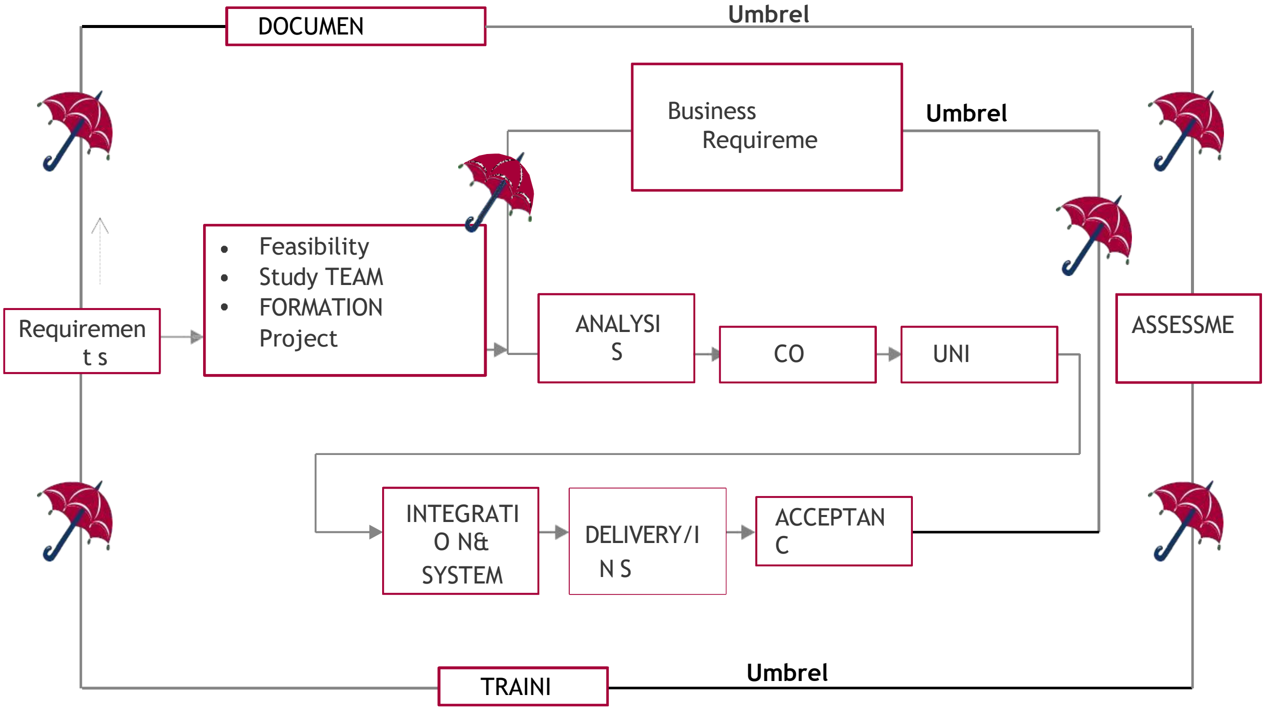


Fig 3.3.1:SDLC Architecture

Requirements Gathering stage:

The requirements gathering process takes as its input the goals identified in the high- level requirements section of the project plan. Each goal will be refined into a set of one ormore requirements. These requirements define the major functions of the intended application,define operational data areas and reference data areas, and define the initial data entities. Majorfunctions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirementtitle and textual description.

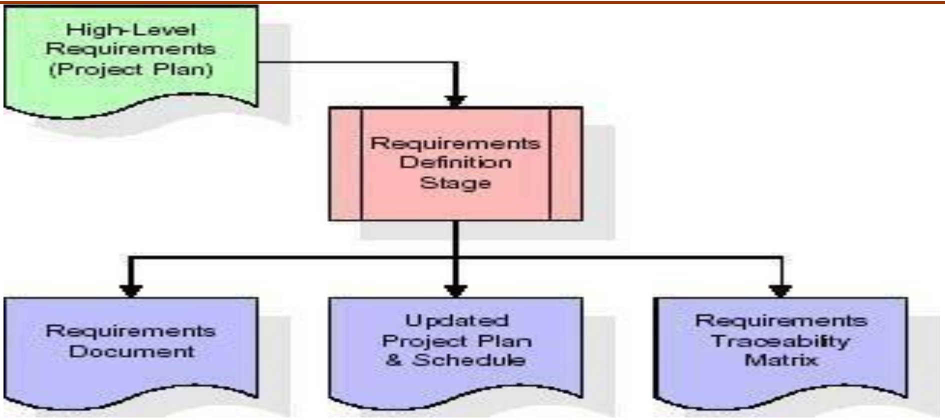

High-Level Requirements (Project Plan)
Requirements Definition Stage
Requirements Document
Updated Project Plan & Schedule
Requirements Traceability Matrix

Fig 3.3.2: Requirementstages in SDLC

These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and

references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in priorstages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage isformally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

♦ Feasibility study is all about identification of problems in a project.

- ♦ No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employeeswho are working for that project.

- ♦ Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

Analysis Stage:

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.

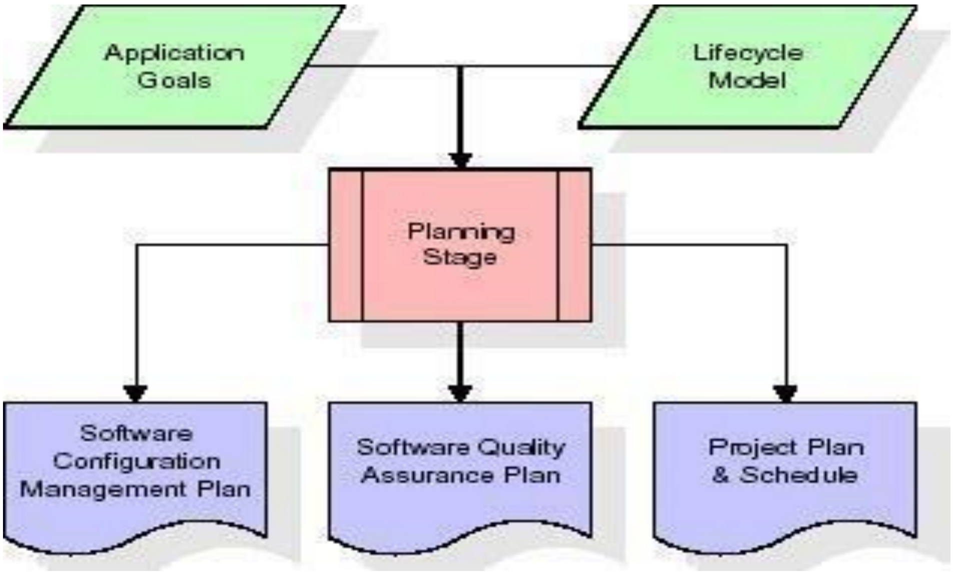

Application Goals
Lifecycle Model
Planning Stage
Software Configuration Management Plan
Software Quality Assurance Plan
Project Plan & Schedule

Fig 3.3.3: Analysis stages in SDLC

The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum informationfor each goal consists of a title and textual description, although additional information and references to external documents may be included.

The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high levelestimates of effort for the out stages.

Designing Stage:

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. Thesedesignelements are intended to describe the software in sufficient detail that skilled programmers maydevelop the software with minimal additional input.

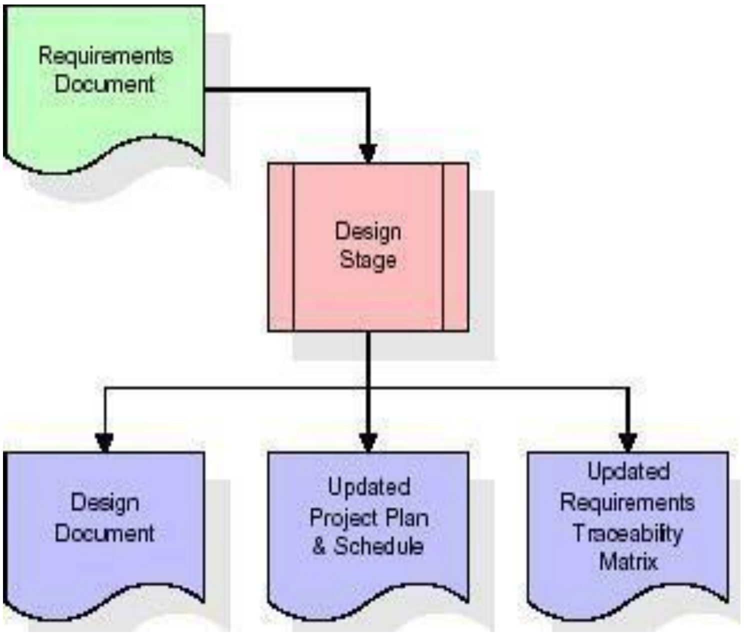

Requirements Document
Design Stage
Design Document
Updated Project Plan & Schedule
Updated Requirements Traceabilty Matrix

Fig 3.3.4: Designing stages in SDLC

When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

Development (Coding) Stage:

The development stage takes as its primary input the design elements described in the approved design document.

For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures andfunctions. Appropriate test cases will bedeveloped for each set of functionallyrelated software artifacts,and an online help systemwill be developed to guideusers in their interactions with the software.

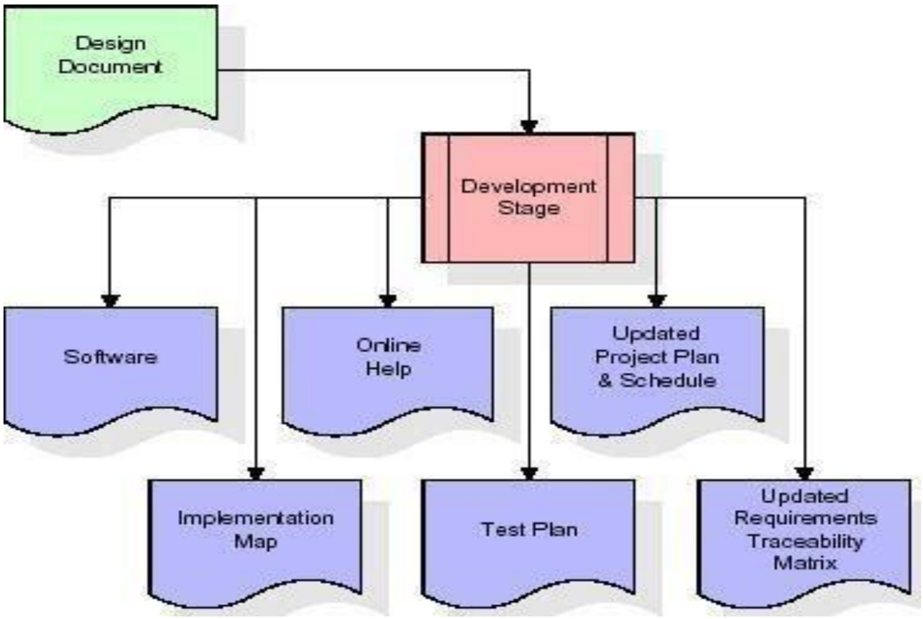


Fig 3.3.5:Development stages in SDLC

The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

Integration & Test Stage:

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment.At this point, all testcases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

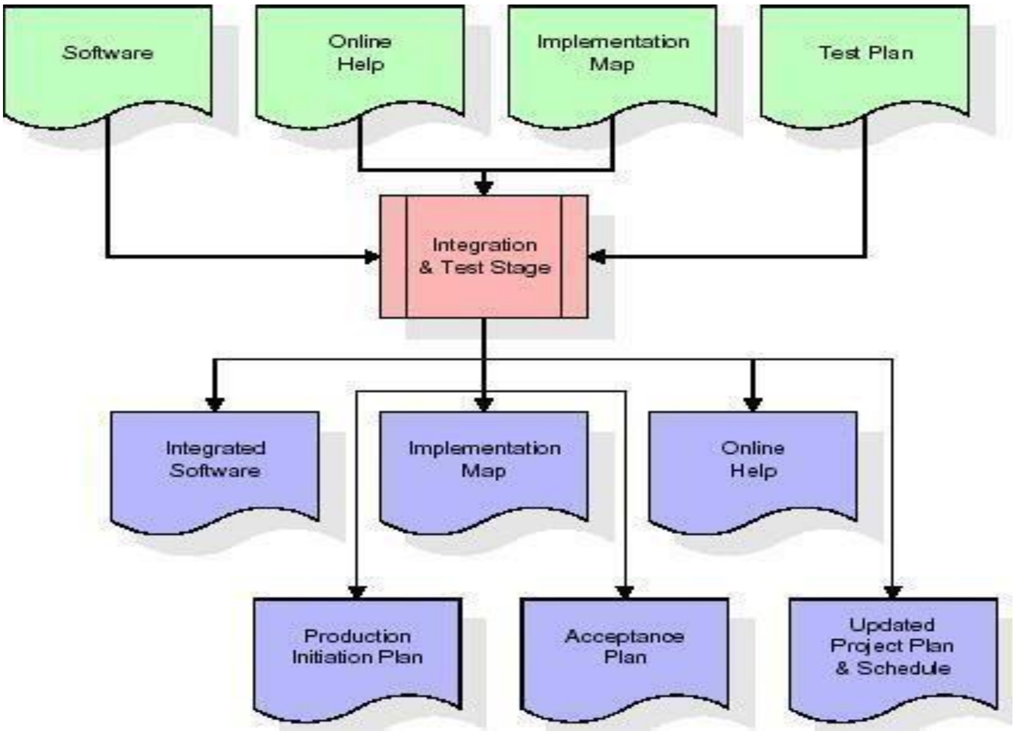


Fig3.3.6: Integration & test stages in SDLC

The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite oftest cases, and an updated project plan.

- **Installation & Acceptance Test:**

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loa ded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successfulexecution of the test suite isa prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load iscorrect and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.

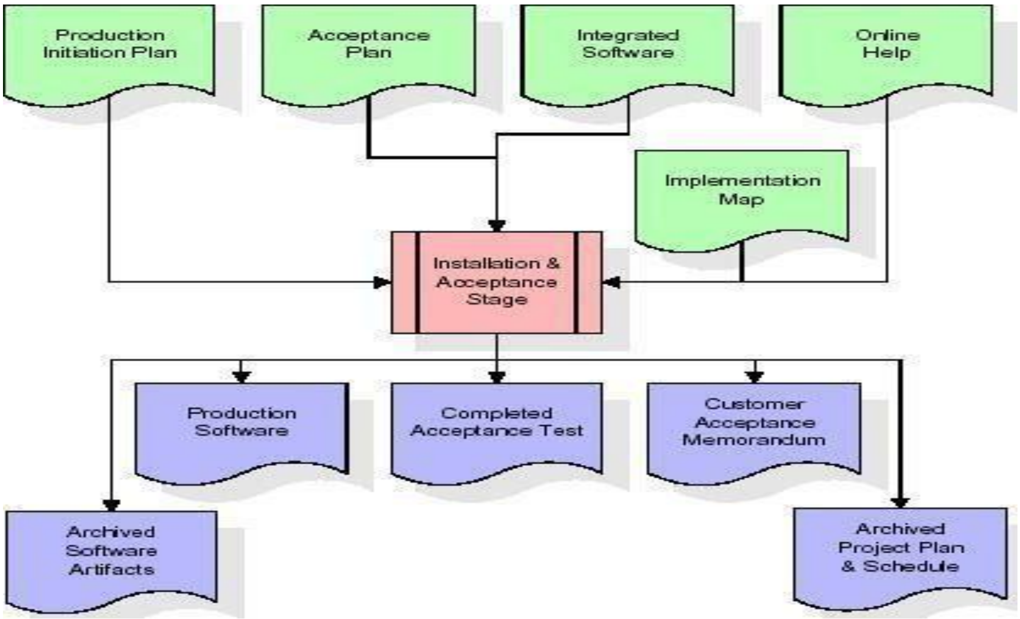

Production Initiation Plan
Acceptance Plan
Integrated Software
Online Help
Implementation Map
Installation & Acceptance Stage
Production Software
Completed Acceptance Test
Customer Acceptance Memorandum
Archived Software Artifacts
Archived Project Plan & Schedule

Fig 3.3.7: Installation & acceptance test in SDLC

The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

Maintenance:

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end,it will be continued so on like an umbrella (no ending point to umbrella sticks).SOFTWARE REQUIREMENT SPECIFICATION:

3.4.1. Overall Description

A Software Requirements Specification (SRS) – a requirements specification for a software system is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non- functional requirements. Nonfunctional requirements are requirements whichimpose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the systems development lifecycle domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

- Business requirements describe in business terms what must be delivered or accomplished to provide value.

- Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)

- Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- **ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must stillbe a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

- **OPERATIONAL FEASIBILITY**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above- mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well- planned design would ensure the optimal utilization of the computer resources and would helpin the improvement of performance status.

- **TECHNICAL FEASIBILITY**

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides aneasy access to .the users. The database's purpose isto create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

3.4.2 External Interface Requirement

User Interface

The user interface of this system is a user friendly python Graphical User Interface.

Hardware Interfaces

The interaction between the user and the console is achieved through python capabilities.

Software Interfaces

The required software is python.

SYSTEM REQUIREMENT:

HARDWARE REQUIREMENTS:

- Processor - Intel i3(min)
- Speed - 2.4 GHz
- RAM - 8 GB (min)
- Hard Disk - 500 GB

SOFTWARE REQUIREMENTS:

- Operating System - Windows10(min)
- Programming Language - Python

CHAPTER-4

SYSTEM STUDY

4.SYSTEM DESIGN

UML Diagram:

The Unified Modelling Language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

User Model View

I. This view represents the system from the users perspective.

II. The analysis representation describes a usage scenario from the end-users perspective.

Structural Model view

I. In this model the data and functionality are arrived from inside the system.

II. This model view models the static structures.

III. Behavioural Model View

It represents the dynamic of behavioural as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

Implementation Model View

In this the structural and behavioural as parts of the system are represented as they are to be built.

Environmental Model View

In this the structural and behavioural aspects of the environment in which the system is to be implemented are represented.

SYSTEM DESIGN

Class Diagram:

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for datamodeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

- The upper part holds the name of the class
- The middle part contains the attributes of the class
- The bottom part gives the methods or operations the class can take or undertake

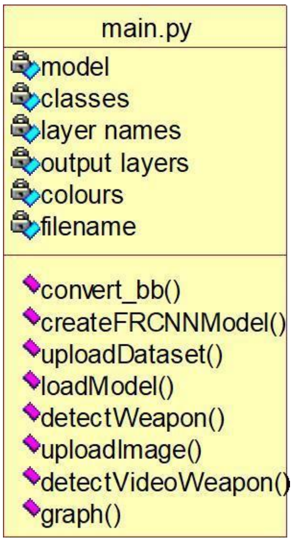


Fig 4.1: Class diagram

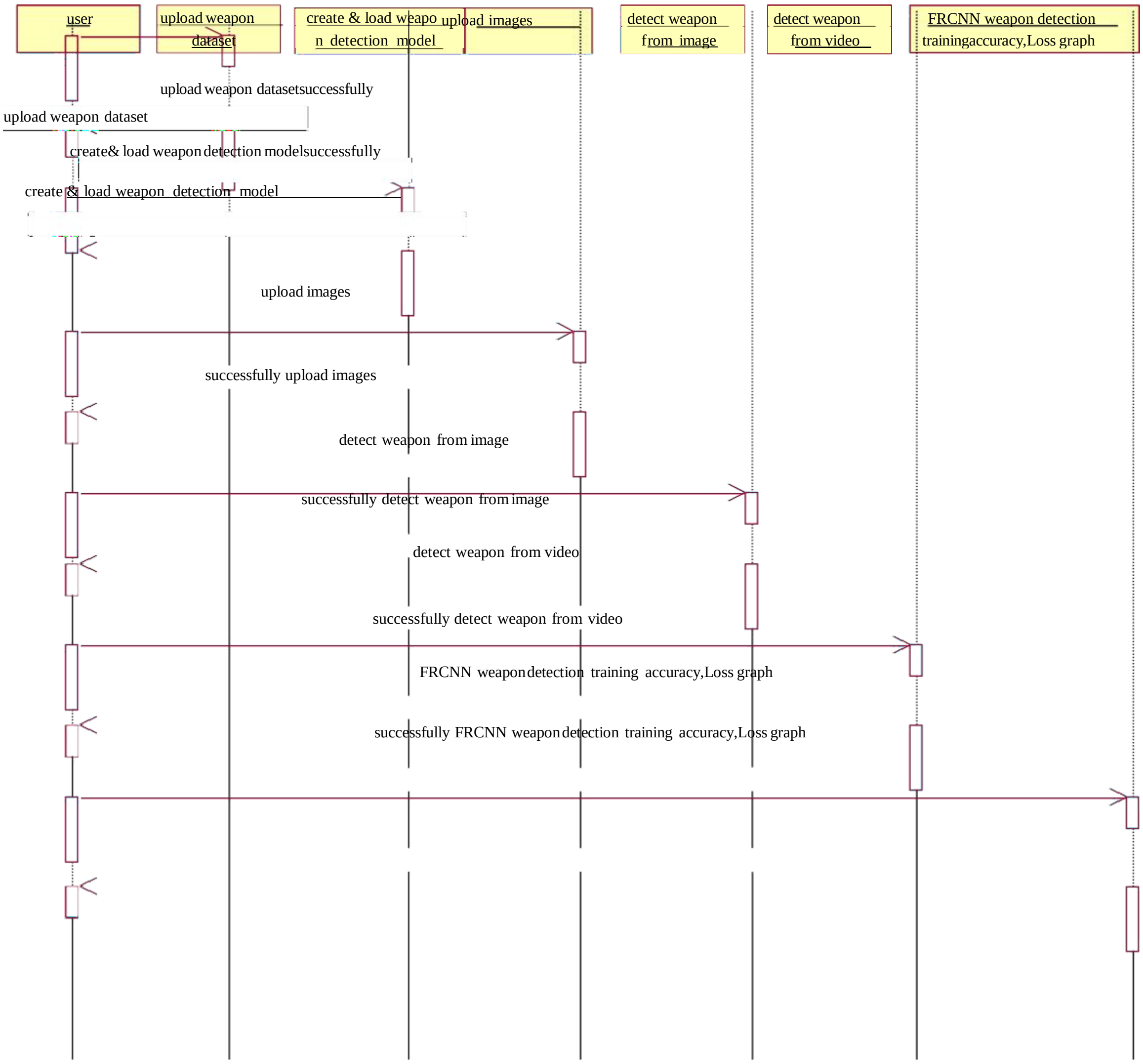
Use case Diagram:

A **use case diagram** at its simplest is a representation of a user's interactionwith the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system.

This type of diagram is typically used in conjunction with the textual use case andwill often be accompanied by other types of diagrams as well.

Sequence diagram:

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of 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 thefunctionality 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 event diagrams, event scenarios, and timing diagrams.


user
upload weapon dataset
create & load weapon detection model
upload images
detect weapon from image
detect weapon from video
FRCNN weapon detection trainingaccuracy,Loss graph
upload weapon datasetsuccessfully
upload weapon dataset
create& load weapon detection modelsuccessfully
create & load weapon detection model
upload images
successfully upload images
detect weapon from image
successfully detect weapon from image
detect weapon from video
successfully detect weapon from video
FRCNN weapondetection training accuracy,Loss graph
successfully FRCNN weapondetection training accuracy,Loss graph

Collaboration diagram:

A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behaviour of a system

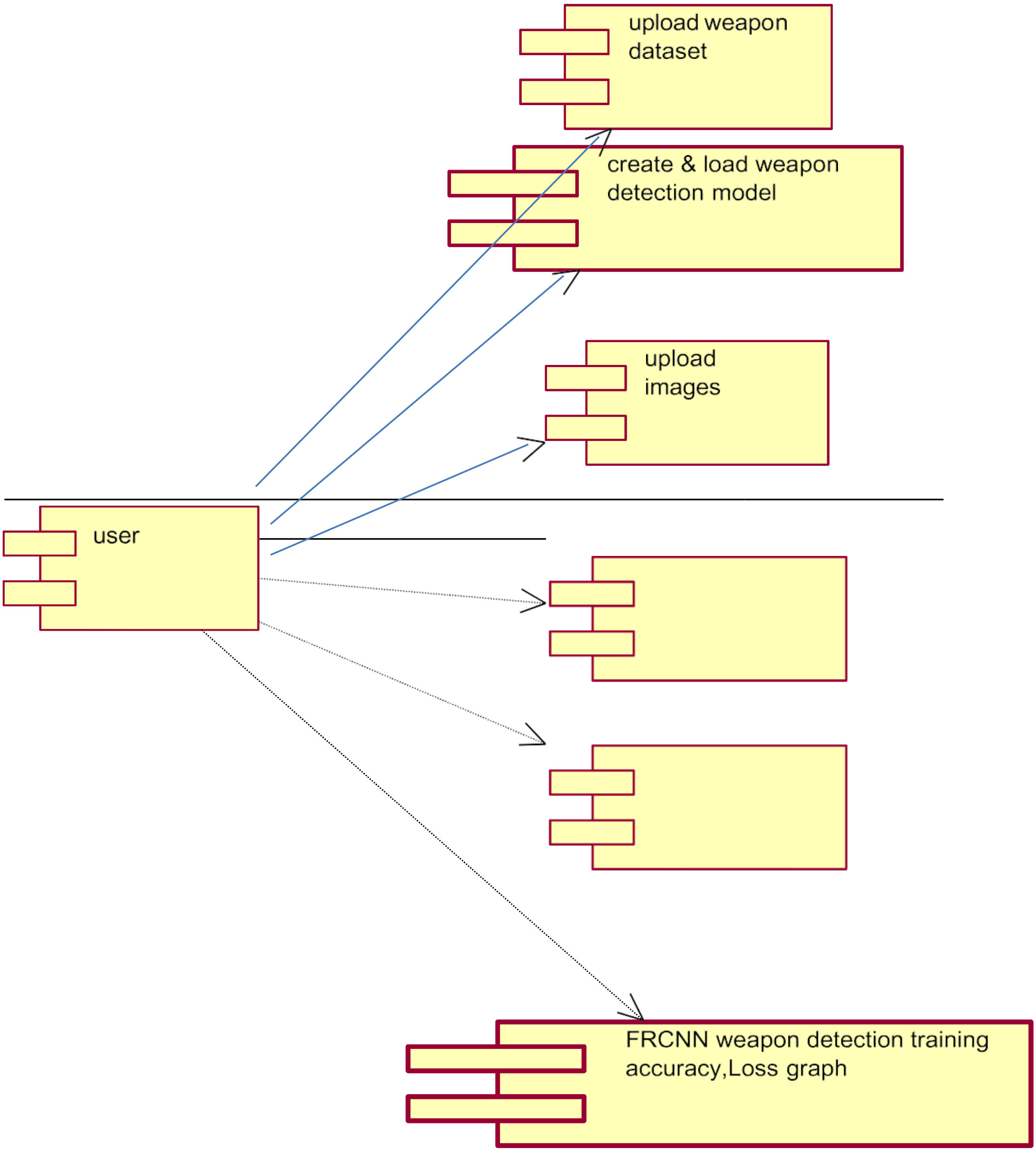

upload weapon dataset
create & load weapon detection model
upload images
user
FRCNN weapon detection training accuracy,Loss graph

Fig 4.3: Collaboration diagram

Component Diagram:

In the Unified Modelling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.

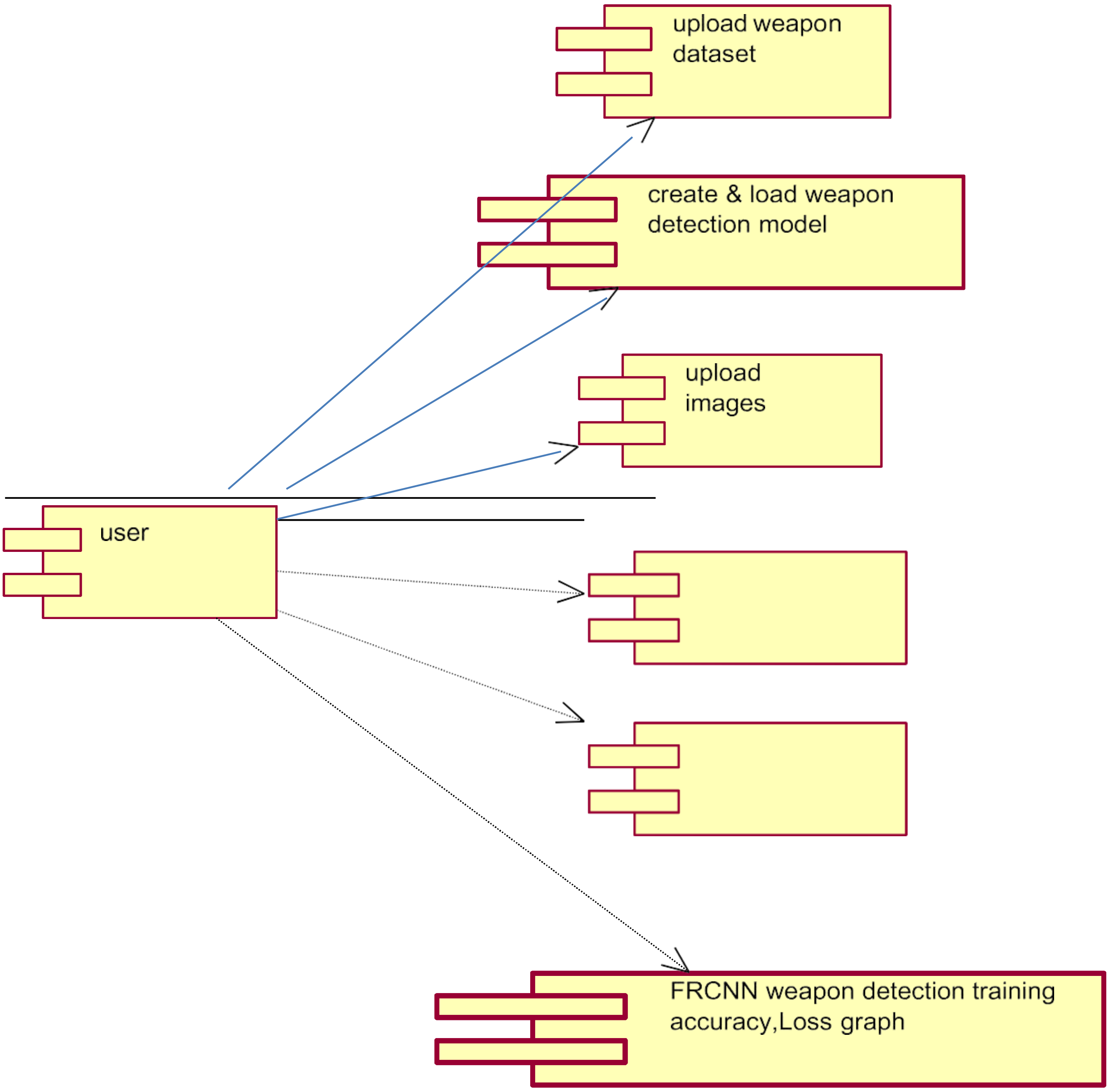

upload weapon dataset
create & load weapon detection model
upload images
user
FRCNN weapon detection training accuracy,Loss graph

Fig 4.4: Component diagram

Deployment Diagram:

A **deployment diagram** in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear asrectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in

a deployment diagram may conceptually represent multiple physical nodes, suchas a cluster of database servers.

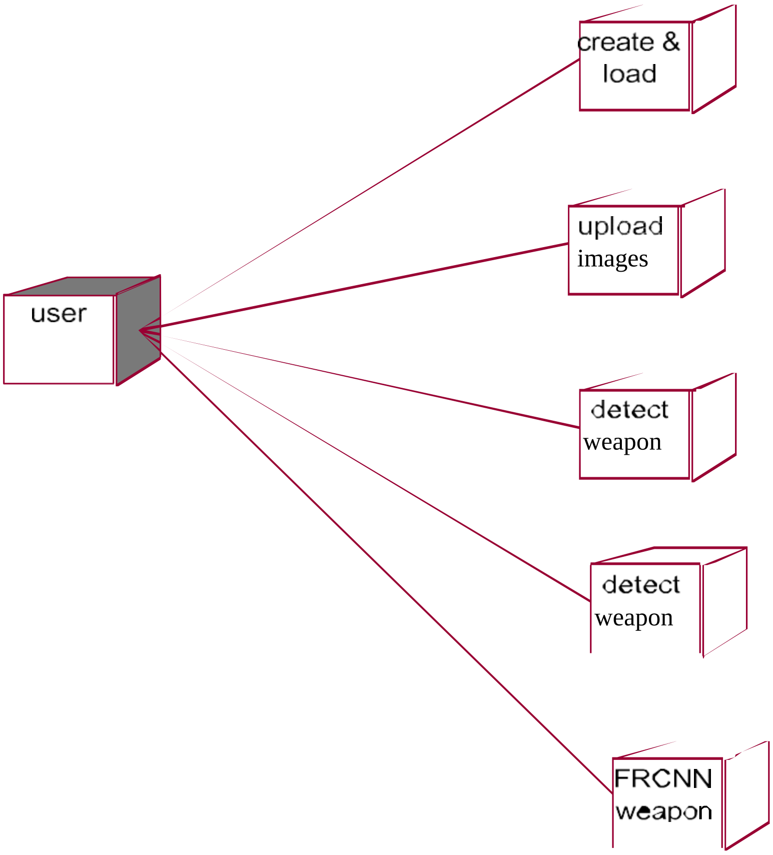


Fig 4.5: Deployment diagram

Activity Diagram:

Activity diagram is another important diagram in UML to describe dynamicaspects of the system. It is basically a flow chart to represent the flow form one activity to another activity.

The activity can be described as an operation of the system. So the control flow isdrawn from one operation to another. This flow can be sequential, branched or concurrent

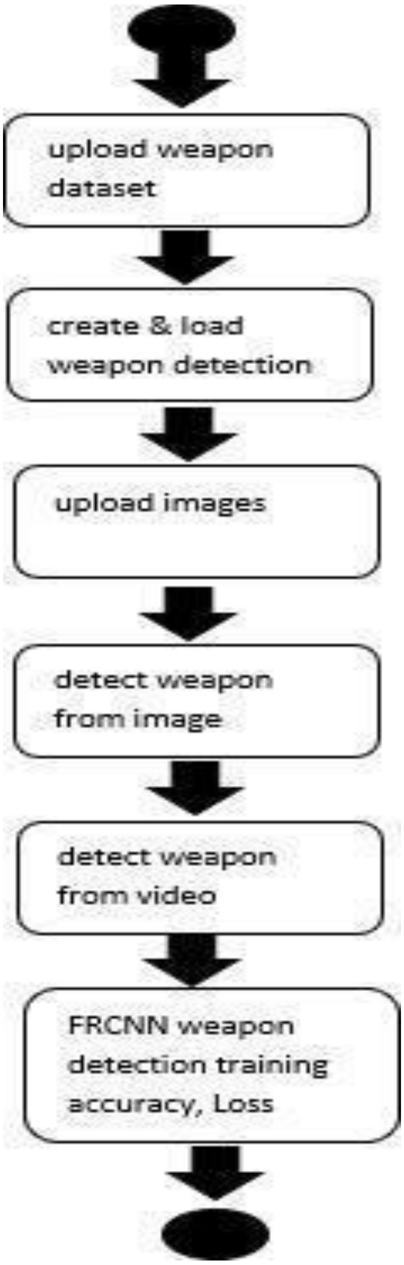


Fig 4.6: Activity diagram

Data Flow Diagram:

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of anybusiness function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to- use, free downloadable diagramming tools. A DFD is a model for constructing andanalyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and theend results

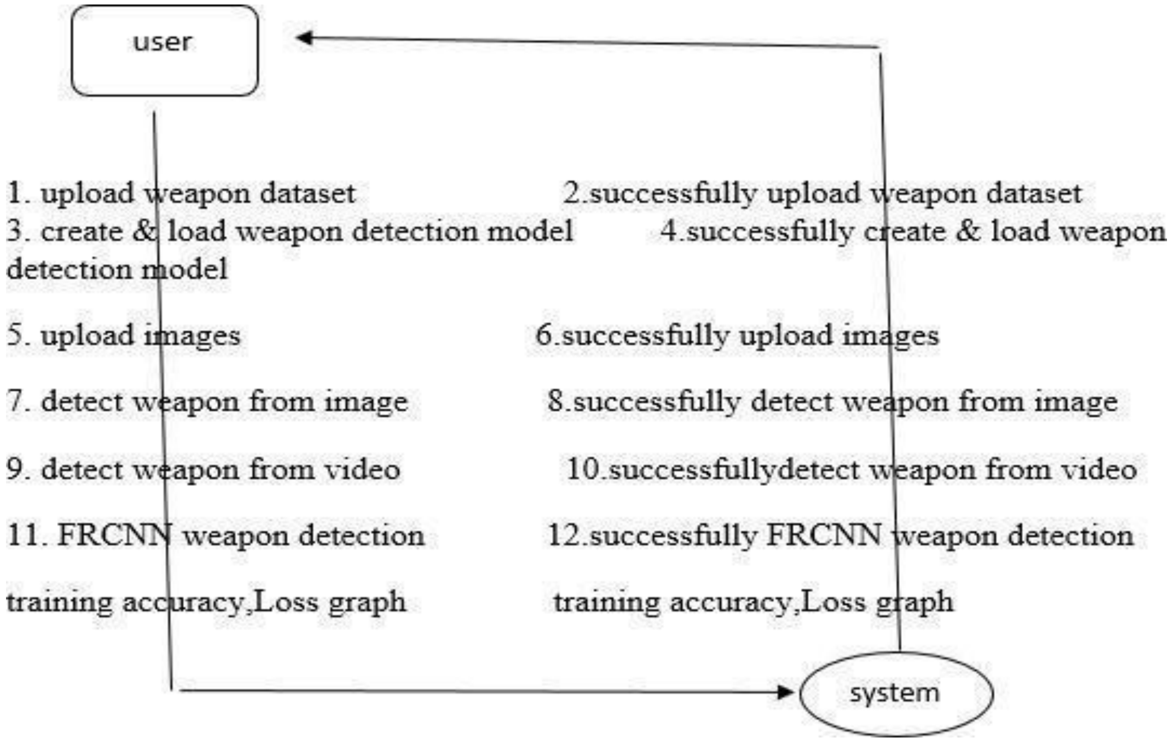

user
1. upload weapon dataset
2.successfully upload weapon dataset
3. create & load weapon detection model
4.successfully create & load weapon detection model
5. upload images
6.successfully upload images
7. detect weapon from image
8.successfully detect weapon from image
9. detect weapon from video
10.successfullydetect weapon from video
11. FRCNN weapon detection training accuracy,Loss graph
12.successfully FRCNN weapon detection training accuracy,Loss graph
system

Fig 4.7:Data flow diagram

CHAPTER-5

IMPLEMENTATION

5.IMPLEMETATION

5.1 Python

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of thelanguage is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on thesyntax.

History of Python:

Python is a fairly old language created by Guido Van Rossum. The design began in the late 1980s and was first released in February 1991.

Why Python was created?

In late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group. He wanted to use an interpreted language like ABC (ABC has simple easy-to- understand syntax) that could access the Amoeba system calls. So, he decided to create a language that was extensible. This led to design of a new language which was later named Python.

Why the name Python?

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's FlyingCircus".

Features of Python:

A simple language which is easier to learn

Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming funand allows you to focus on the solution rather than syntax.

If you are a newbie, it's a great choice to start your journey with Python.

Free and open-source

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software's written in it, you can even make changes to the Python's source code.

Python has a large community constantly improving it in each iteration.

Portability

You can move Python programs from one platform to another, and run it without any changes.

It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

Extensible and Embeddable

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code.

This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

A high-level, interpreted language

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.

Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower- level operations.

Large standard libraries to solve common tasks

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using import MySQLdb .

Standard libraries in Python are well tested and used by hundreds of people. So you can besure that it won't break your application.

Object-oriented

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively.

With OOP, you are able to divide these complex problems into smaller sets by creatingobjects.

Applications of Python:

1. Simple Elegant Syntax

Programming in Python is fun. It's easier to understand and write Python code. Why? The syntax feels natural. Take this source code for an example:

a = 2

b = 3 sum = a +b

print(sum)

2. Not overly strict

You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolon at the end of the statement.

Python enforces you to follow good practices (like proper indentation). These smallthings can make learning much easier for beginners.

3. Expressiveness of the language

Python allows you to write programs having greater functionality with fewer lines of code. Here's a link to the source code of Tic-tac-toe game with a graphical interface and a smart computer opponent in less than 500 lines of code. This is just an example. You will be amazed how much you can do with Python once you learn the basics.

4. Great Community and Support

Python has a large supporting community. There are numerous active forums onlinewhich can be handy if you are stuck.

5.2 Sample

code: from tkinter

```
import * import tkinter

from tkinter import filedialog

from tkinter.filedialog import

askopenfilenameimport numpy as np

import cv2

from keras.utils.np_utils import

to_categoricalfrom keras.layers import

MaxPooling2D

from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense,

Inputfrom keras.models import Model

from keras.optimizers import Adam

from keras.callbacks import ModelCheckpoint

from sklearn.model_selection import

train_test_splitimport pickle

import os
import matplotlib.pyplot as plt

import xml.etree.ElementTree as ET

from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score

main = tkinter.Tk()

main.title("Weapon Detection")

main.geometry("1300x900")

global model, classes, output_layers, colors, filename

X = []

Y = []

bb = []

# Function to normalize bounding boxes

def convert_bb(img, width, height, xmin, ymin, xmax, ymax):

    conv_x = (128. / width)

    conv_y = (128. / height)

    height = ymax * conv_y

    width = xmax * conv_x

    x = max(xmin * conv_x, 0)

    y = max(ymin * conv_y, 0)

    x = x / 128

    y = y / 128
    width = width / 128

    height = height / 128

    return x, y, width, height

def createFRCNNModel():

    global X, Y, bb

    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)

    input_img = Input(shape=(128, 128, 3))

    x = Conv2D(32, (3, 3), padding='same', activation='relu')(input_img)

    x = Conv2D(32, (3, 3), padding='same', activation='relu')(x)

    x = MaxPooling2D((2, 2))(x)

    x = Conv2D(64, (3, 3), padding='same', activation='relu')(x)

    x = Conv2D(64, (3, 3), padding='same', activation='relu')(x)

    x = MaxPooling2D((2, 2))(x)

    x = Flatten()(x)

    x = Dense(512, activation='relu')(x)

    x = Dense(512, activation='relu')(x)

    x_bb = Dense(4, name='bb')(x)

    x_class = Dense(2, activation='softmax', name='class')(x)

    frcnn_model = Model([input_img], [x_bb, x_class])

    frcnn_model.compile(Adam(lr=0.001), loss=['mse', 'categorical_crossentropy'], metrics=['accuracy'])
    if not os.path.exists("model/frcnn_model_weights.hdf5"):
        model_check_point = ModelCheckpoint(filepath='model/frcnn_model_weights.hdf5', verbose=1,
save_best_only=True)
        hist = frcnn_model.fit(X, [bb, Y], batch_size=32, epochs=10, validation_split=0.2,
callbacks=[model_check_point])
        with open('model/history.pckl', 'wb') as f:
            pickle.dump(hist.history, f)
    else:
        frcnn_model.load_weights("model/frcnn_model_weights.hdf5")
    predict = frcnn_model.predict(X_test)
    predict = np.argmax(predict[1], axis=1)
    test = np.argmax(y_test, axis=1)
    p = precision_score(test, predict, average='macro') * 100
    r = recall_score(test, predict, average='macro') * 100
    f = f1_score(test, predict, average='macro') * 100
    a = accuracy_score(test, predict) * 100
    text.insert(END, 'FRCNN Model Accuracy : ' + str(a) + "\n")
    text.insert(END, 'FRCNN Model Precision : ' + str(p) + "\n")
    text.insert(END, 'FRCNN Model Recall    : ' + str(r) + "\n")
    text.insert(END, 'FRCNN Model FMeasure  : ' + str(f) + "\n\n")
    text.update_idletasks()

def uploadDataset():

    global X, Y, bb

    filename = filedialog.askdirectory(initialdir="Dataset/annotations")

    bb = []  # Initialize bb outside the else block

    if os.path.exists('model/X.txt.npy'):  # if dataset images already processed then load it

        X = np.load('model/X.txt.npy') # load X images data

        Y = np.load('model/Y.txt.npy')  # load weapon class label

        bb = np.load('model/bb.txt.npy')  # load bounding boxes

        Y = to_categorical(Y)

    else:

        for root, dirs, directory in os.walk('Dataset/annotations/xmls'):  # if not processed images then loop

all annotation files with bounding boxes

            for j in range(len(directory)):

                tree = ET.parse('Dataset/annotations/xmls/' + directory[j])

                root = tree.getroot()

                img_name = str(root.find('filename').text)  # convert to string

                for item in root.findall('object'):

                    name = item.find('name').text  # read class id

                    xmin = int(item.find('bndbox/xmin').text)  # read all bounding box coordinates
```

```
          ymin = int(item.find('bndbox/ymin').text)

          xmax = int(item.find('bndbox/xmax').text)

          ymax = int(item.find('bndbox/ymax').text)

          img = cv2.imread("Dataset/images/" + img_name)  # read image path from xml

          height, width, channel = img.shape

          img = cv2.resize(img, (128, 128)) # Resize image

          img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

          x, y, width, height = convert_bb(img, width, height, xmin, ymin, xmax, ymax)  # normalized

bounding boxes

          Y.append(0) # add weapon label to Y array

          bb.append([x, y, width, height])  # add bounding boxes

          X.append(img)

    X = np.asarray(X)  # convert array to numpy format

    Y = np.asarray(Y)

    bb = np.asarray(bb)

    np.save('model/X.txt', X)  # save all processed images

    np.save('model/Y.txt', Y)

    np.save('model/bb.txt', bb) text.insert(END, "Dataset Loaded\n")

  text.insert(END, "Total images found in dataset : " + str(X.shape[0])+ "\n\n")
```

```
def loadModel():
    text.delete('1.0', END)
    global model, classes, layer_names, output_layers, colors
    model = cv2.dnn.readNet("model/frcn.checkpoints", "model/frcn_config.cfg")
    classes = ['Weapon']
    layer_names = model.getLayerNames()
    output_layer_indices = model.getUnconnectedOutLayers()  # Get the indices directly
    if isinstance(output_layer_indices, int):  # Check if it's an integer
        output_layer_indices = [output_layer_indices] # Convert to list for uniform handling
    output_layers = [layer_names[i[0] - 1] for i in output_layer_indices]  # Adjust index here
    colors = np.random.uniform(0, 255, size=(len(classes), 3))
    createFRCNNModel()
    text.insert(END, "Weapon Detection Model Loaded\n")
def detectWeapon():
    global model, classes, output_layers, colors, filename
    filename = filedialog.askopenfilename(initialdir="testImages")
    img = cv2.imread(filename)
    height, width, channels = img.shape
    blob = cv2.dnn.blobFromImage(img, 0.00392, (416, 416), (0, 0, 0), True, crop=False)
```

```
model.setInput(blob)

outs = model.forward(output_layers)

class_ids = []

confidences = []

boxes = []

score = []

for out in outs:

    for detection in out:

        scores = detection[5:]

        class_id = np.argmax(scores)

        confidence = scores[class_id]

        scr = np.amax(scores)

        if confidence > 0.5:

            center_x = int(detection[0] * width)

            center_y = int(detection[1] * height)

            w = int(detection[2] * width)

            h = int(detection[3] * height)

            x = int(center_x - w / 2)

            y = int(center_y - h / 2)

            score.append(scr)
            boxes.append([x, y, w, h])
            confidences.append(float(confidence))
            class_ids.append(class_id)
indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)
if indexes == 0:
    text.insert(END, "Weapon detected in image\n")
flag = 0
font = cv2.FONT_HERSHEY_PLAIN
for i in range(len(boxes)):
    if i in indexes:
        x, y, w, h = boxes[i]
        print(str(class_ids[i]) + " " + str(score[i]))
        label = str(classes[class_ids[i]])
        color = colors[class_ids[i]]
        cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)
        cv2.putText(img, label, (x, y + 30), font, 3, color, 3)
        flag = 1
if flag == 0:
    cv2.putText(img, "No weapon Detected", (10, 25), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 0, 0), 2)
    cv2.imshow("Image", img)

    cv2.waitKey(0)

def uploadImage():

    global filename

    text.delete('1.0', END)

    filename = askopenfilename(initialdir="testImages")

    text.insert(END, filename + " loaded\n")

def detectVideoWeapon():

    global model, classes, output_layers, colors, filename

    filename = filedialog.askopenfilename(initialdir="Videos")

    cap = cv2.VideoCapture(filename)

    while True:

        _, img = cap.read()

        if img is None:

            break

        detectWeaponFromFrame(img)

        key = cv2.waitKey(1)

        if key == ord('q'):

            break
    cap.release()

    cv2.destroyAllWindows()

def detectFromWebcam():

    global model, classes, output_layers, colors, filename

    cap = cv2.VideoCapture(0) # Open the webcam

    while True:

        _, img = cap.read()

        if img is None:

            break

        detectWeaponFromFrame(img)  # Call the function to detect weapons

        key = cv2.waitKey(1)

        if key == ord('q'):

            break

    cap.release()

    cv2.destroyAllWindows()

def detectWeaponFromFrame(img):

    global model, classes, output_layers, colors

    height, width, channels = img.shape

    blob = cv2.dnn.blobFromImage(img, 0.00392, (416, 416), (0, 0, 0), True, crop=False)

    model.setInput(blob)
```

```
outs = model.forward(output_layers)

class_ids = []

confidences = []

boxes = []

for out in outs:

    for detection in out:

        scores = detection[5:]

        class_id = np.argmax(scores)

        confidence = scores[class_id]

        if confidence > 0.5:

            # Object detected

            center_x = int(detection[0] * width)

            center_y = int(detection[1] * height)

            w = int(detection[2] * width)

            h = int(detection[3] * height)

            # Rectangle coordinates

            x = int(center_x - w / 2)

            y = int(center_y - h / 2)

            boxes.append([x, y, w, h])

            confidences.append(float(confidence))
```

```
                class_ids.append(class_id)

    indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)

    if indexes == 0:

        print("No weapon detected in frame")

    font = cv2.FONT_HERSHEY_PLAIN

    for i in range(len(boxes)):

        if i in indexes:

            x, y, w, h = boxes[i]

            label = str(classes[class_ids[i]])

            color = colors[class_ids[i]]

            cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)

            cv2.putText(img, label, (x, y + 30), font, 3, color, 3)

    cv2.imshow("Image", img)

def graph():

    f = open('model/history.pckl', 'rb')

    data = pickle.load(f)

    f.close()

    accuracy = data['class_accuracy']

    loss = data['class_loss']
    fig, axs = plt.subplots(1, 2, figsize=(12, 6))
    axs[0].plot(accuracy, 'ro-')  # Remove the color parameter here
    axs[0].set_title("FRCNN Accuracy Graph")
    axs[0].set_xlabel('Epochs')
    axs[0].set_ylabel('Accuracy')
    axs[1].plot(loss, 'ro-')  # Remove the color parameter here
    axs[1].set_title("FRCNN Loss Graph")
    axs[1].set_xlabel('Epochs')
    axs[1].set_ylabel('Loss')
    plt.show()
font = ('times', 16, 'bold')
title = Label(main, text='Weapon Detection', anchor=W, justify=LEFT)
title.config(bg='black', fg='white')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0, y=5)
font1 = ('times', 13, 'bold')
uploadDatasetButton = Button(main, text="Upload Weapon Dataset", command=uploadDataset)
uploadDatasetButton.place(x=50, y=100)
uploadDatasetButton.config(font=font1)
```

```
loadButton = Button(main, text="Generate & Load Weapon Detection Model", command=loadModel)

loadButton.place(x=300, y=100)

loadButton.config(font=font1)

uploadButton = Button(main, text="Upload Image", command=uploadImage)

uploadButton.place(x=50, y=150)

uploadButton.config(font=font1)

detectButton = Button(main, text="Detect Weapon from Image", command=detectWeapon)

detectButton.place(x=50, y=200)

detectButton.config(font=font1)

videoButton = Button(main, text="Detect Weapon from Video", command=detectVideoWeapon)

videoButton.place(x=50, y=250)

videoButton.config(font=font1)

webcamButton = Button(main, text="Detect Weapon from Webcam", command=detectFromWebcam)

webcamButton.place(x=50, y=300)

webcamButton.config(font=font1)

graphButton = Button(main, text="FRCNN Weapon Detection Training Accuracy-Loss Graph",

command=graph)

graphButton.place(x=50, y=350)

graphButton.config(font=font1)
```

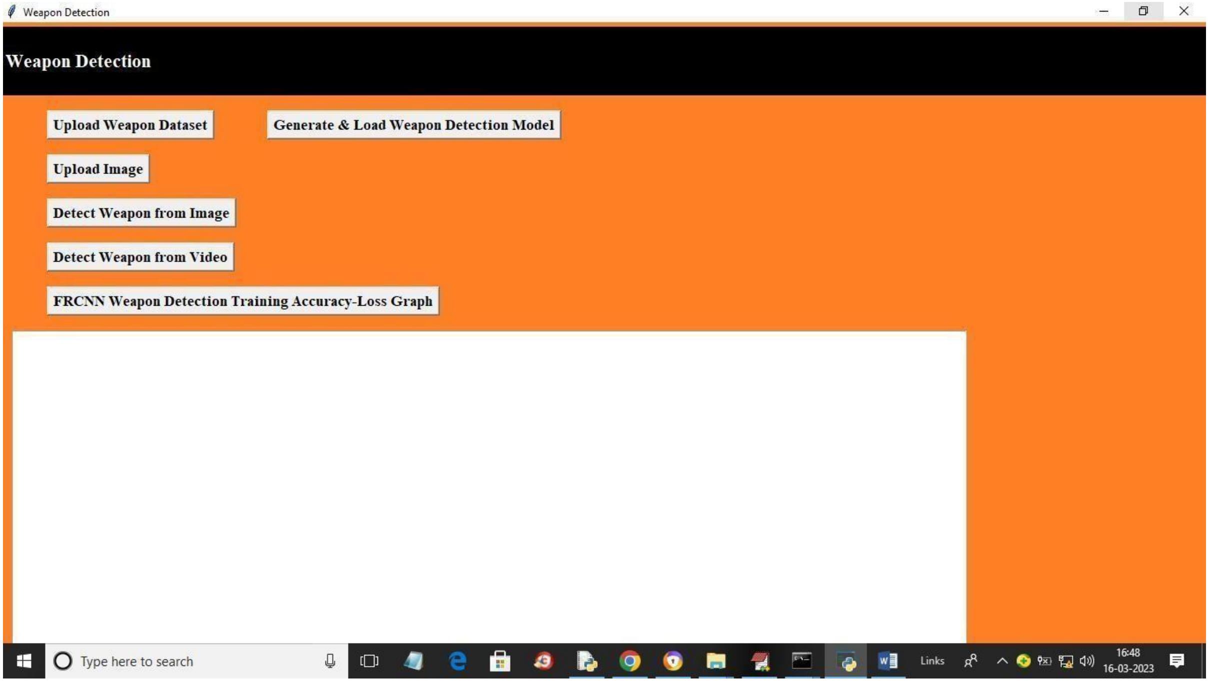
```
text = Text(main, height=20, width=120)

scroll = Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=10, y=400)

text.config(font=font1)

main.config(bg='chocolate1')

main.mainloop()
```

CHAPTER-6

SCREEN SHOTS

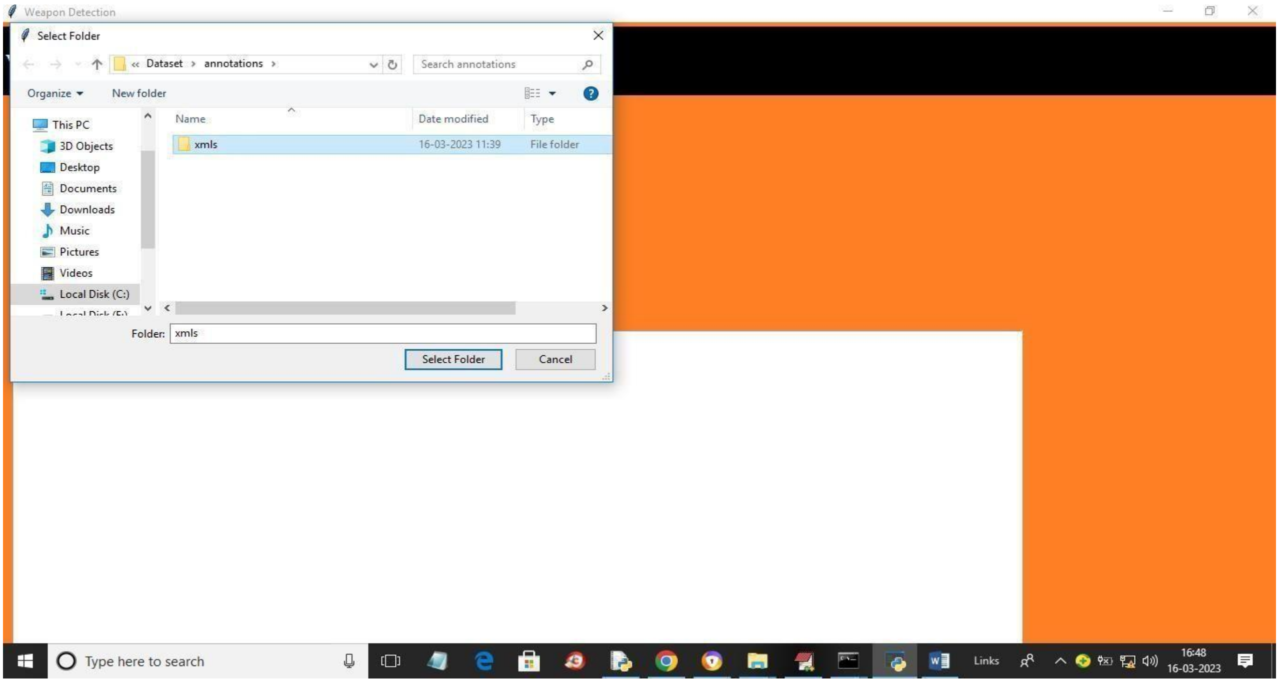
6. SCREEN SHOTS

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



6.1 Upload Weapon Dataset

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



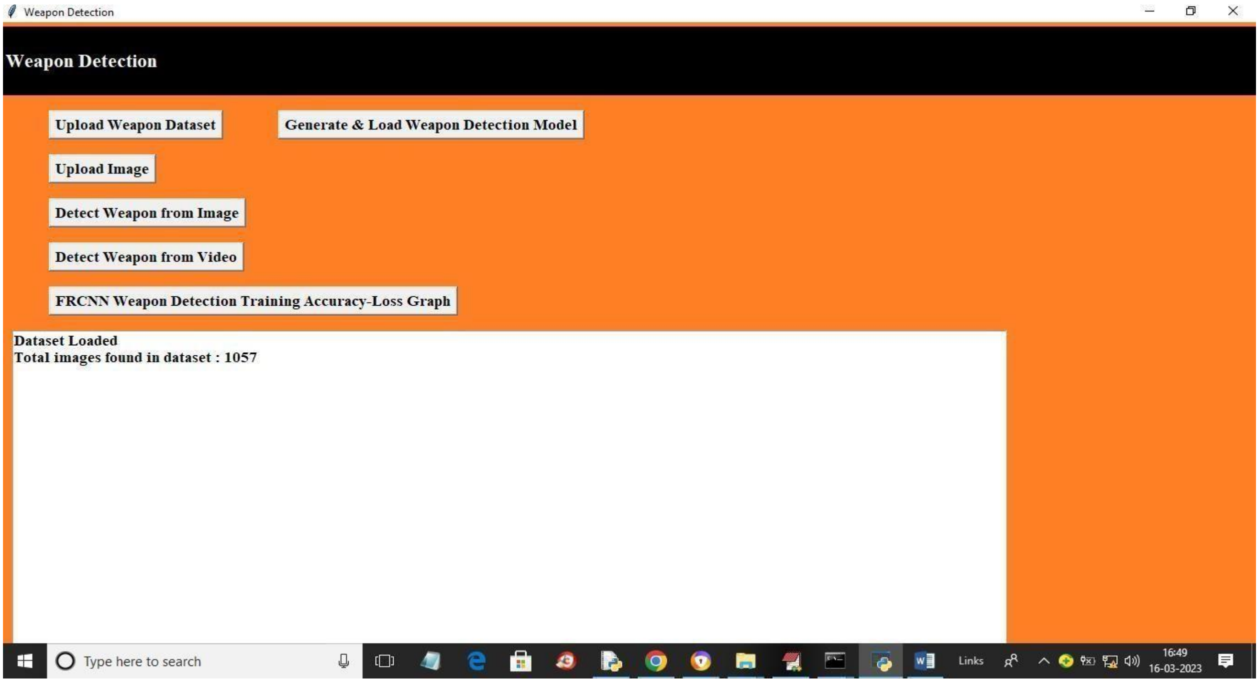
Name	Date modified	Type
xmls	16-03-2023 11:39	File folder

Folder: xmls

6.2 Selecting the Folder

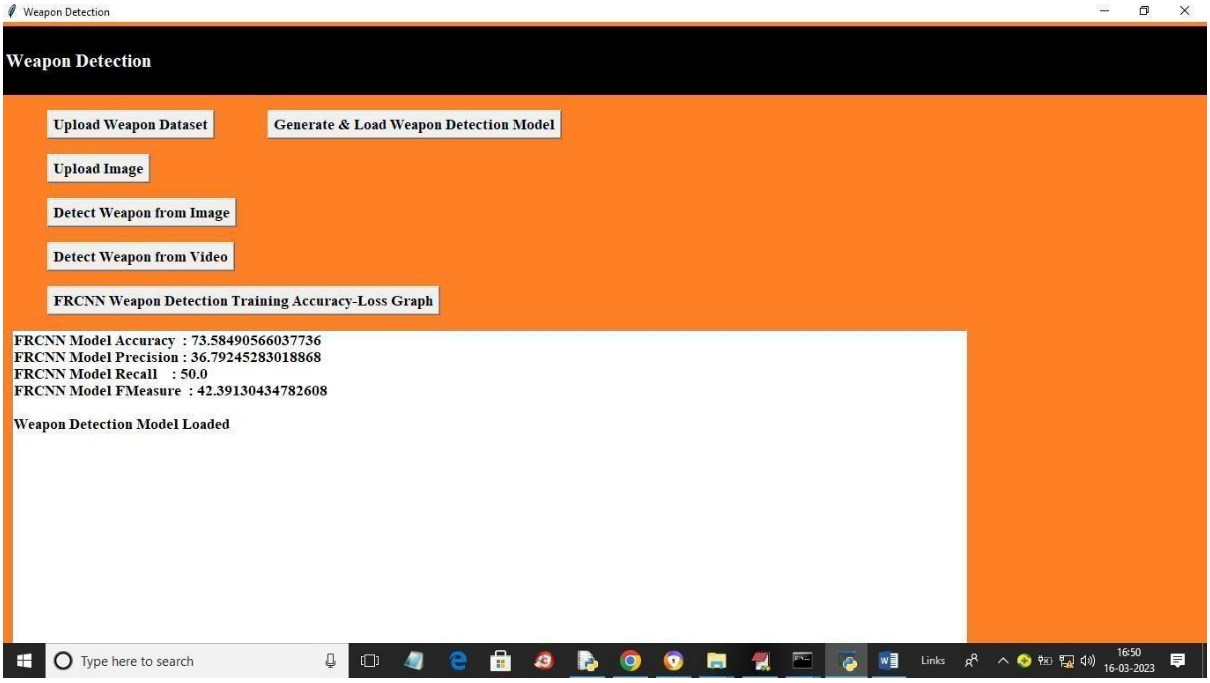
In above screen selecting and uploading entire XML folder with annotating bounding boxes and images path and then click on ‘Select Folder’ button to loaddataset and get below output

6.1 Generate & Load Weapon Detection Model



```
Dataset Loaded
Total images found in dataset : 1057
```

In above screen we can see 1057 weapon images loaded from dataset and now click on ‘Generate & Load Weapon Detection Model’ button to train FRCNN model and get below output

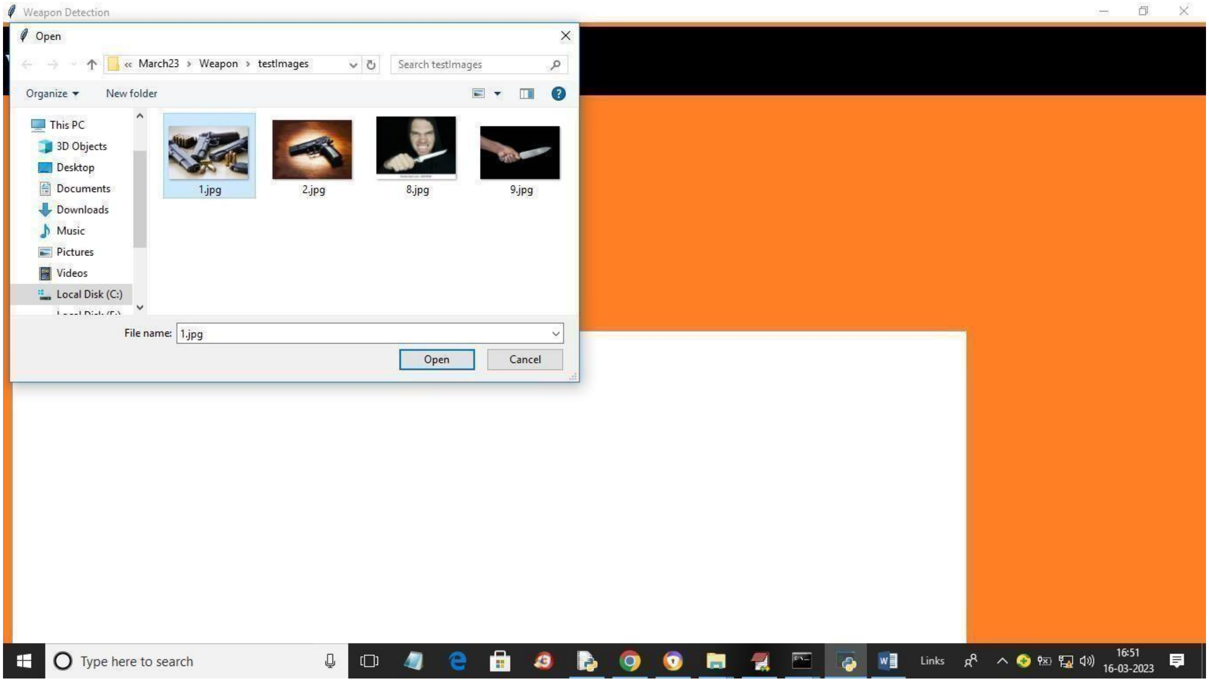


```
FRCNN Model Accuracy  : 73.58490566037736
FRCNN Model Precision : 36.79245283018868
FRCNN Model Recall    : 50.0
FRCNN Model FMeasure  : 42.39130434782608

Weapon Detection Model Loaded
```

6.2 Uploading Images

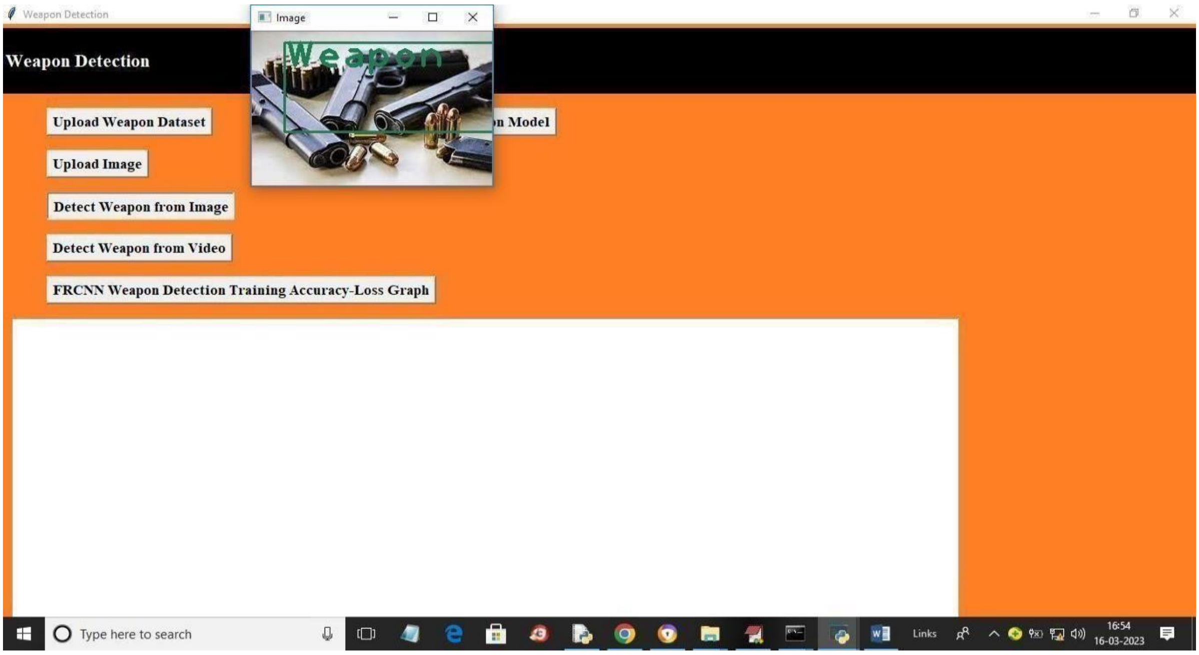
In above screen FRCNN model loaded and we got it accuracy as 73% and we can see other

metrics also and now click on 'Upload Image' button to upload image and get below output



6.3 Detecting Weapon From Images

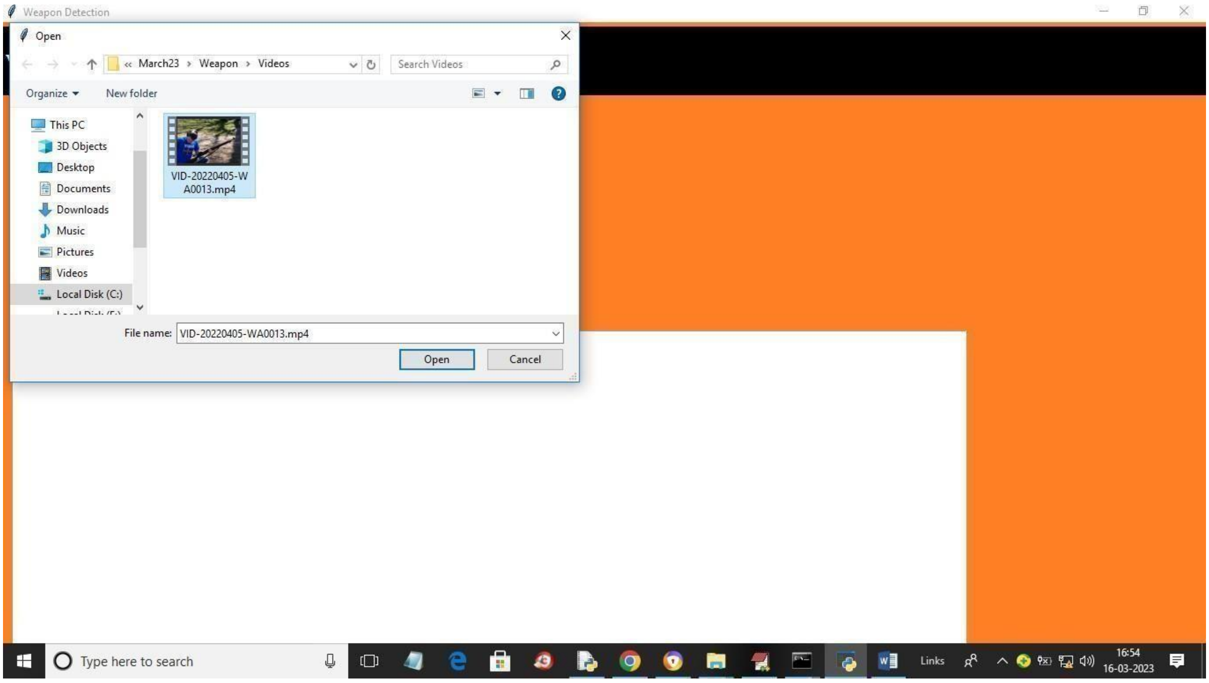
In above screen selecting and uploading image and then click on 'Open' button and 'Detect Weapon from Image' button to get below output



6.4 Detecting Weapon From Videos

In above screen in image weapon is detected and now similarly click on 'Detect

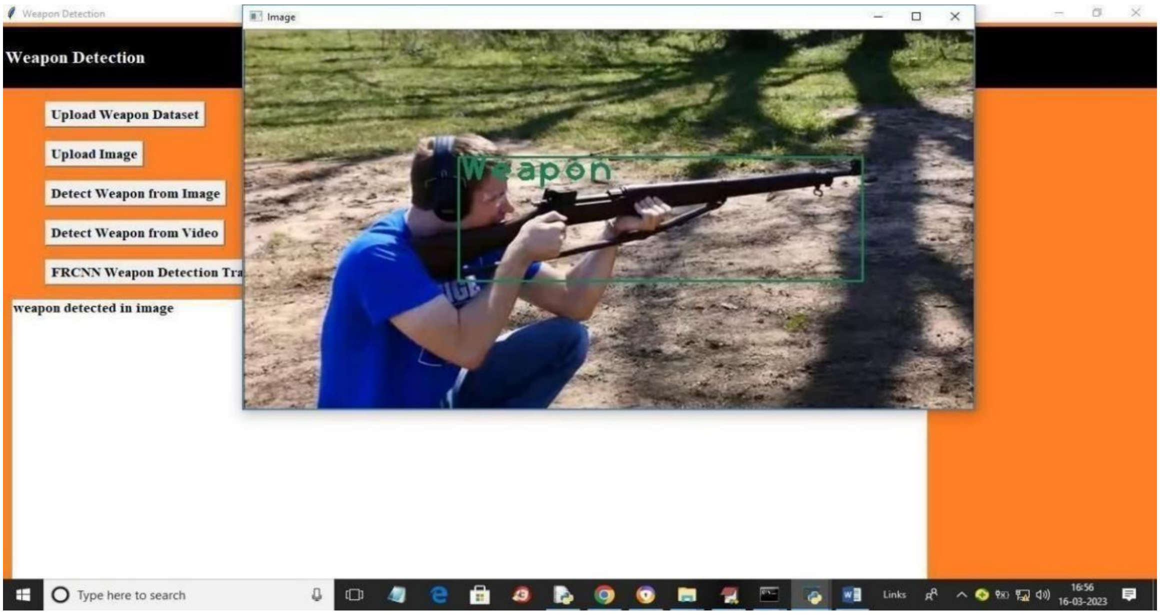
Weapon from Video' button to upload video file and get below output



File name: VID-20220405-WA0013.mp4

6.5 Uploading MP4Video Files

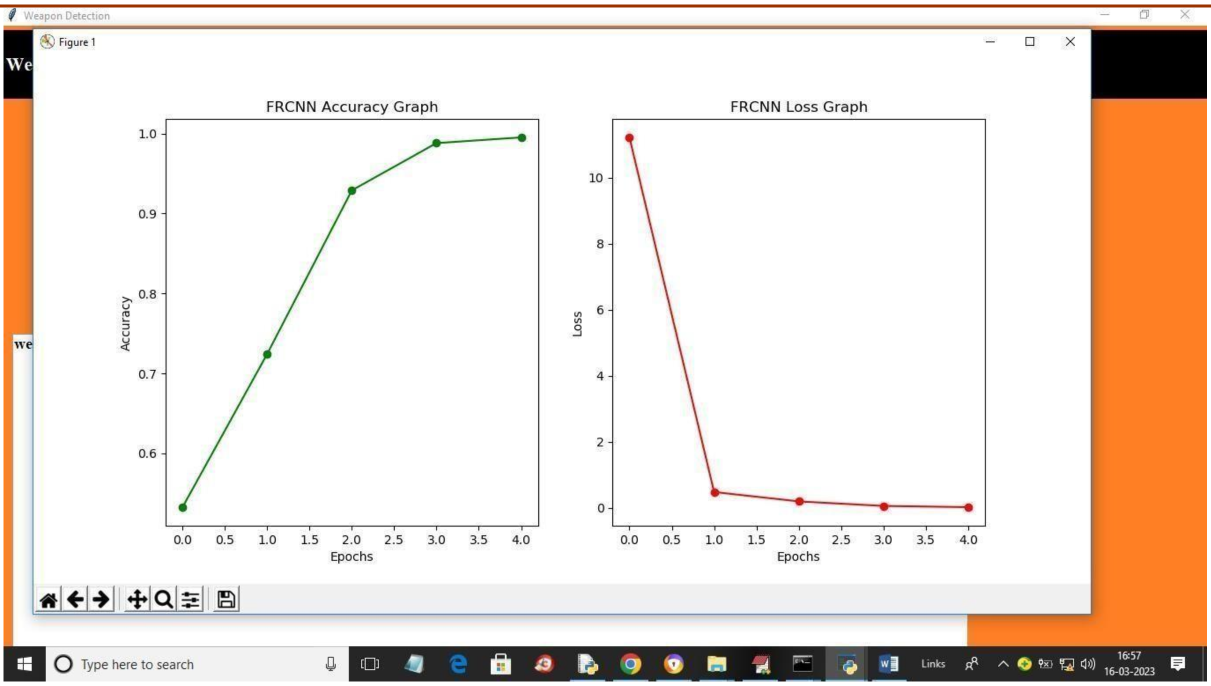
In above screen selecting and uploading MP4 video file and then click on 'Open' button to start video playing and if your system fast then video will play faster otherwise slow



weapon detected in image

6.6 Detecting the Weapon

In above screen you can see video start playing and we can see detection outputalso and now click on 'FRCNN Weapon Detection Training Accuracy-Loss Graph' button to get below output



6.7 FRCNN Weapon Detection Training Accuracy-Loss Graph

In above training graphs x-axis represents training EPOCH and y-axis represents accuracy and loss values and green line represents accuracy and red line represents LOSS and with each increasing epoch accuracy got increase and loss got decrease. Similarly you can upload and test other images.

Test cases:

Test Case Id	Test Case Name	Test Case Desc.	Test Steps			Test Case Status	Test Priority
			Step	**Expected**	**Actual**		
01	upload weapon dataset	Upload the weapon dataset	If weapon Dataset is not uploaded we cannot do anything	we cannot do any further operations	we can do further operations	High	High
02	create & load weapon detection model	Verify model created or not	If model not is not done	we cannot do any further operations	we can do further operations	High	High
03	upload images	Verify images are uploaded or not	If uploading is not done	We cannot run operation	We can Run the Operation	High	High
04	detect weapon from images	Verify the weapon detection	If detection is not done	We cannot run operation	We can Run the Operation	High	High
05	detect weapon from video	Verify the weapon detection from video	If detection is not done	We cannot run another operation	We can Run the Operation	High	High
06	FRCNN	Cheque The accuarcy and loss	If graph is plottted or not	We cannot run operation	We Can run the oper ation	High	High

CHAPTER-7

CONCLUSION

7. CONCLUSION

SSD and Faster RCNN algorithms are simulated for pre labeled and self-created image dataset for weapon (gun) detection. Both the algorithms are efficient and give good results but their application in real time is based on a tradeoff between speed and accuracy. In terms of speed, SSD algorithm gives better speed with 0.736 s/frame. Whereas Faster RCNN gives speed 1.606s/frame, which is poor compared to SSD. With respect to accuracy, Faster RCNN gives better accuracy of 84.6%. Whereas SSD gives an accuracy of 73.8%, which is poor compared to faster RCNN.SSD provided real time detection due to faster speed but Faster RCNN provided superior accuracy. Further, it can be implemented for larger datasets by training us ing GPUs and high-end DSP and FPGA kits.

8. REFERENCES

[1] Wei Liu et al., “SSD: Single Shot MultiBox Detector”, European Conference on Conputer Vision, Volume 169, pp 20-31 Sep. 2017.

[2] D. Erhan et al., “Scalable Object Detection Using Deep Neural Networks,” IEEE Conference on Computer Vision and Pattern Recognition(CVPR),2014.

[3] Ruben J Franklin et.al., “Anomaly Detection in Videos for Video Surveillance Applications Using Neural Networks,” International Conference on Inventive Systems and Control,2020.

[4] H R Rohit et.al., “A Review of Artificial Intelligence Methods for Data Science and Data Analytics: Applications and Research Challenges,”2018 2nd InternationalConference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), 2018.

[5] Abhiraj Biswas et. al., “Classification of Objects in Video Records using NeuralNetwork Framework,” International conference on Smart Systems and Inventive Technology,2018.

[6] Pallavi Raj et. al.,“Simulation and Performance Analysis of Feature Extraction and Matching Algorithms for Image Processing Applications” IEEE International Conference on Intelligent Sustainable Systems,2019.

[7] Mohanaet .al.,“Simulation of Object Detection Algorithms for Video Survillance Applications”, International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud),2018.

[8] Yojan Chitkara et. al.,“Background Modelling techniques for foreground detection and Tracking using Gaussian Mixture model” International Conference on Computing Methodologies and Communication,2019.

[9] Rubner et.al, “A metric for distributions with applications to image databases”, International Conference on Computer Vision,2016.

[10] N. Jain et.al., “Performance Analysis of Object Detection and Tracking Algorithms for Traffic Surveillance Applications using Neural Networks,” 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), 2019.

[11] A. Glowacz et.al., “Visual Detection of Knives in Security Applications using Active Appearance Model”,Multimedia Tools Applications, 2015.

[12] S. Pankanti et.al.,“Robust abandoned object detection using regionlevel

analysis,"International Conference on Image Processing,2011.

MUKt SHABd JOURNAL

Mukt Shabd Journal

UGC CARE GROUP - I JOURNAL

ISSN NO : 2347-3150 / web www.shabdbooks.com / e-mail : submitmsj@gmail.com

ज्ञान-विज्ञान विमुक्तये UGC APPROVED

IMPACT FACTORS 4.6

DOI:09.0014.MSJ

crossref member CROSSREF.ORG THE CITATION LINKING BACKBONE

CERTIFICATE OF PUBLICATION

This is to certify that the paper entitled

" EFFECTIVE USAGE OF ARTIFICIAL INTELLIGENCEAND DEEP LEARNING FOR IMPLEMENTING WEAPON DETECTION FOR SECURITY APPLICATIONS "

Authored by

Mr. Niranjan Babu Thanikanti

From

SVR Engineering College,Nandyal.

Has been published in

MUKT SHABD JOURNAL, VOLUME XIII, ISSUE IV, APRIL- 2024

Sumit Ganguly
Editor-In-Chief
MSJ
www.shabdbooks.com

Mukt Shabd Journal

UGC CARE GROUP - I JOURNAL

ISSN NO : 2347-3150 / web www.shabdbooks.com / e-mail : submitmsj@gmail.com

ज्ञान-विज्ञान विमुक्तये
UGC
APPROVED

IMPACT FACTORS 4.6

DOI:09.0014.MSJ
crossref member
CROSSREF.ORG
THE CITATION LINKING BACKBONE

CERTIFICATE OF PUBLICATION

This is to certify that the paper entitled

" EFFECTIVE USAGE OF ARTIFICIAL INTELLIGENCEAND DEEP LEARNING FOR IMPLEMENTING WEAPON DETECTION FOR SECURITY APPLICATIONS "

Authored by

Shaik Ruheena Tanveer, Mekala Vigneswar Reddy, Vanipenta Shasikala

From

SVR Engineering College,Nandyal.

Has been published in

MUKT SHABD JOURNAL, VOLUME XIII, ISSUE IV, APRIL- 2024

Sumit Ganguly
Editor-In-Chief
MSJ
www.shabdbooks.com

EFFECTIVE USAGE OF ARTIFICIAL INTELLIGENCEAND DEEP LEARNING FOR IMPLEMENTING WEAPON DETECTION FOR SECURITY APPLICATIONS

Mr. Niranjan Babu Thanikanti
Assistant Professor[1],
Shaik Ruheena Tanveer[2], Mekala Vigneswar Reddy[3], Vanipenta Shasikala[4]
Department of CSE(ARTIFICIAL INTELLIGENCE),
SVR Engineering College,Nandyal

ABSTRACT

Because crime tends to spike in busy places or in suspiciously empty ones, security is always an important issue in any field. A number of issues may be addressed via the use of computer vision, particularly in the areas of abnormal detection and monitoring. Intelligence monitoring relies heavily on video surveillance systems that can identify and understand the scene, as well as abnormal occurrences, to meet the rising need for safety, security, and personal property protection. Automatic gun or weapon identification is implemented in this study utilising Faster RCNN and SSD, two techniques based on convolutional neural networks (CNNs). Two kinds of datasets are used in the proposed implementation. There was one dataset with photographs that had already been tagged, and another with images that were to be labelled by hand. Both methods attain high accuracy, and the results are tabulated. However, their practical applicability may depend on the compromise between speed and precision.

INTRODUCTION

Anamoly detection, also known as weapon detection, is the process of spotting anything out of the ordinary in a dataset, such as an occurrence that does not fit the typical pattern or an object that is not part of the regular pattern. A pattern that deviates from a predetermined set of norms is called an anomaly. Thus, the phenomena of interest determines whether there are anomalies. In order to identify examples of different types of things, object detection makes use of extracted features and learning models or algorithms. The accuracy of gun detection and categorization is the main emphasis of the proposed implementation. Also worried about precision, as unintended consequences may ensue from a false warning. Finding a happy medium between precision and speed necessitated picking the correct strategy. The input video is used to extract frames. Prior to object identification, the bounding box is formed and the frame differencing technique is used.

RELATED WORK

“SSD: Single Shot Detector”

We provide a technique that employs just one deep neural network to identify visual objects. Our method, which we call SSD, uses a variety of aspect ratios or scales each feature map position to discretize the output space or bounding boxes into an assortment of default boxes. During the prediction process, the network determines how well each default box fits the form of the item and assigns points based on the existence of each object type. To handle objects of varied sizes naturally, the network also incorporates predictions from numerous map features with different resolutions. Compared to approaches that rely on object proposals, SSD is more straightforward since it consolidates all processing into a single network, doing away with the need to generate proposals and the following phases of pixel or feature resampling. This facilitates the training process and the integration of SSD into systems requiring a detection component. With an integrated structure for training and inference, SSD outperforms approaches that use an extra object proposal phase in terms of speed and accuracy, according to experimental findings on the the PASCAL VOC, COCO, or ILSVRC datasets. **“Scalable Object Detection Using Deep Neural Networks,”**

The ImageNet Massive Visual Identification Challenge (ILSVRC-2012) is only one of many image recognition benchmarks where deep convolution neural network models have lately attained state-of-the-art performance. A network that can identify different types of objects in a picture and then forecast their bounding boxes and confidence scores was the most successful model in the localization sub-task. This kind of model is great for capturing the surrounding environment of an item in a picture, but it is foolish to think it can manage more than one instance of a single object without producing the same amount of outputs. Our proposed detection technique is based on a saliency-inspired neural network. It provides a collection of class-agnostic bounding boxes and assigns a score to each box based on its probability of holding an item of interest. At its most advanced levels, the model is able to generalise across classes and automatically manages an adjustable number of instances per class. We achieve competitive recognition results on VOC2007 and ILSVRC2012 utilising a modest number of artificial neural network evaluations and using just the top 10 predicted locations for each picture.

“Anomaly Detection in Videos for Video Surveillance Applications Using Neural Networks,”

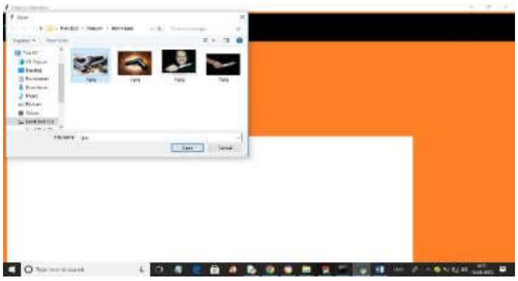
Anomaly detection in videos for video surveillance applications using neural networks refers to the process of automatically identifying irregular or unexpected events within a continuous stream of video data captured by surveillance cameras. Traditional video surveillance systems often rely on predefined rules or handcrafted features to detect anomalies, which can be limited in their ability to adapt to complex scenes or subtle anomalies. An approach to pattern recognition known as anomaly detection may quickly identify outliers—patterns that deviate significantly from the norm. Several plausible outliers may be caught on camera by surveillance systems. Separating the process into its component parts—video labelers, processing of images, and activity detection—allows for more accurate recognition of anomalies in video surveillance. As a consequence, when it comes to real-time situations, anomaly detection in videos in video surveillance applications guarantees outcomes.

“Performance Analysis of Object Detection and Tracking Algorithms for Traffic Surveillance Applications using Neural Networks,”

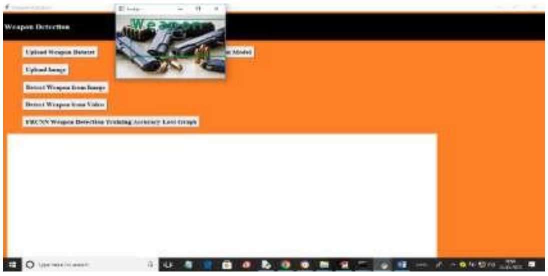
Using the principles of convolution layers, one object detection was successfully executed. Among the many layers that make up a neural network are the input, hidden, and output layers. When doing single object detection, the dataset used is that of the on-road car dataset. Heavy, Auto, and Light are the three picture categories that make up this dataset. The pictures in the collection have different levels of light. The day, evening, and night datasets have all had their performance metrics computed. The You Only Looked Once (YOLOv3) method has been used for multiple item identification. One deep neural network with convolution is used in this method; the input is divided into a cell grid, and each cell immediately predicts a border box and classes the item. For the purpose of multiple object detection, the KITTI dataset is used. There are a total of 80 classes in it, but just five—car, bus, truck, motorbike, and train—have been viewed for this project. Vehicle tracking was further integrated using the principles of Multiple Object Detection. Multi object detection was applied to the first frames of the video, and the item was tracked in subsequent frames by using its centroid location. The object identification phase was built using the YOLOv3 method in Python and OpenCV.

METHODOLOGY

In this research, we provide an implementation of the based SS D and quicker RCNN algorithms for automated weapon identification. Two kinds of datasets are used in the proposed implementation. There was one dataset with photographs that had already been tagged, and another with images that were to be labelled by hand. Both methods attain high accuracy, and the results are tabulated. However, their practical applicability may depend on the trade-off among speed and precision.

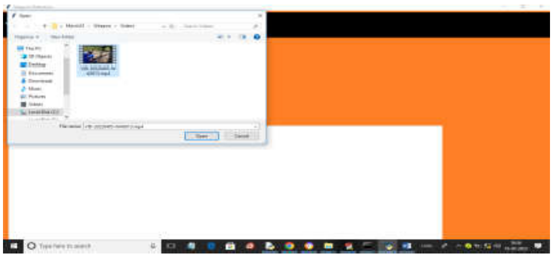


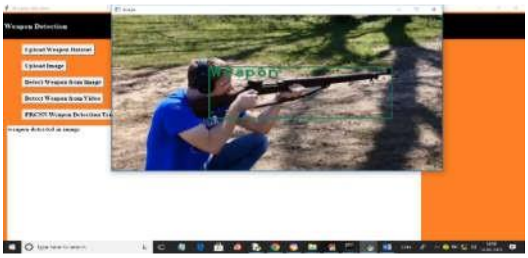
RESULTS AND DISCUSSION



After choosing and uploading a picture to the previous page, you may get the following result by clicking the "Open" and "Detect Weapon from Image" buttons.

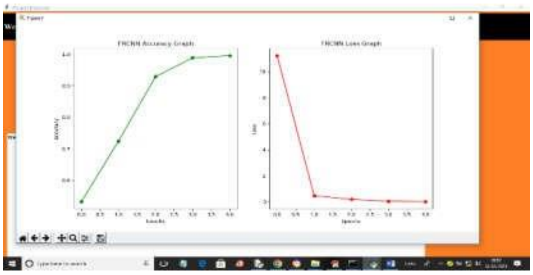
You can see that a weapon has been recognised in the picture up there. To do the same with a video file, just click the "Detect Weapon from Video" option. The results will be shown below.





weapon detected in image

If your system is fast, the video will play quickly; otherwise, it will take some time. To begin playing the video, pick and upload an MP4 file, and then click the "Open" button.You can see the video starting to play and the detection output on the previous screen. To view the output below, click on the accuracy, achieving an impressive 84.6%. In contrast to the quicker RCNN, SSD only achieves a 73.8% accuracy rate.The quicker speed of SSD allowed for real-time detection, "FRCNN Weapon Identification Training Accuracy-Loss Graph" button.



The x-axis of the training graph above shows the epochs of training, while the y-axis shows the values of accuracy and loss; the green line shows the accuracy and the red line shows the loss. As the epochs grow, the accuracy increases and the loss decreases. Other photographs may also be uploaded and tested in a similar manner.

CONCLUSION

For the purpose of weapon (gun) identification, we simulate the SSD and quicker RCNN algorithms on both pre-labeled and self-created picture datasets. There is a compromise between rapidity and precision when using both methods in real time, but they are efficient and provide decent results. While comparing algorithms, the SSD approach provides faster results, at 0.736 s/frame. In contrast, SSD achieves a speed of 1.606s/frame, whereas Faster RCNN falls far short. Faster RCNN outperforms the other methods in terms.while the better accuracy of quicker RCNN was evident. Additionally, it may be trained using GPUs и elite DSP and FPGA packages, allowing it to be deployed for bigger datasets.

REFERENCES

[1] Wei Liu et al., “SSD: Single Shot MultiBox Detector”, European Conference on Conputer Vision, Volume 169, pp 20-31 Sep. 2017.

[2] D. Erhan et al., “Scalable Object Detection Using Deep Neural Networks,” IEEE Conference on Computer Vision and Pattern

Recognition (CVPR),2014.

[3] Ruben J Franklin et.al., “Anomaly Detection in Videos for Video Surveillance Applications Using Neural Networks,” International Conference on Inventive Systems and Control,2020.

[4] H R Rohit et.al., “A Review of Artificial Intelligence Methods for Data Science and Data Analytics: Applications and Research Challenges,”2018 2nd International onference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), 2018.

[5] Abhiraj Biswas et. al., “Classification of Objects in Video Records using Neural Network Framework,” International conference on Smart Systems and Inventive Technology,2018.

[6] Pallavi Raj et. al.,“Simulation and Performance Analysis of Feature Extraction and Matching Algorithms for Image Processing Applications” IEEE International Conference on Intelligent Sustainable Systems,2019. [7] Mohana et.al., “Simulation of Object Detection Algorithms for Video Survillance Applications”, International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud),2018.

[8] Yojan Chitkara et. al.,“Background Modelling techniques for foreground detection and Tracking using Gaussian Mixture model” International Conference on Computing Methodologies and Communication,2019.

[9] Rubner et.al, “A metric for distributions with applications to image databases”, International Conference on Computer Vision,2016.

[10] N. Jain et.al., “Performance Analysis of Object Detection and Tracking Algorithms for Traffic Surveillance Applications using Neural Networks,” 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), 2019.

[11] A. Glowacz et.al., “Visual Detection of Knives in Security Applications using Active Appearance Model”,Multimedia Tools Applications, 2015.

[12] S. Pankanti et.al.,“Robust abandoned object detection using regionlevel analysis, “International Conference on Image Processing,2011.

[13] Ayush Jain et.al.,“Survey on Edge Computing - Key Technology in Retail Industry” International Conference on Intelligent Computing and Control Systems,2019.

[14] Mohana et.al., Performance Evaluation of Background Modeling Methods for Object Detection and Tracking,” International Conference on Inventive Systems and Control,2020.

[15] J. Wang et.al., “Detecting static objects in busy scenes”, Technical Report TR99-1730, Department of Computer Science, Cornell University, 2014.