# Weapon Identification using YOLO V5 Algorithm

**Divyamsh Reddy A[1], M Poojitha[2], G Puspalalitha[3], M Vishnu Vardhan Reddy[4], K Ashok Kumar[5], G Nithya Sree[6]**

Demoera Consulting Pvt Ltd
Email: *divyamsh2001[at]gmail.com*

**Abstract:** ***Rapid advancement of computer vision technologies has led to significant progress in the field of object detection, with applications ranging from surveillance to autonomous vehicles. One critical application is the identification of weapons in real-time scenarios, such as security monitoring and law enforcement. This paper presents a novel approach for weapon identification using the YOLOv5 (You Only Look Once version 5)algorithm, a state-of-the-art real-time object detection model. Traditional methods of weapon identification often rely on manual intervention or limited rule-based systems, which can be time-consuming and prone to errors. In contrast, the proposed method harnesses the powerof deep learning to automate the detection process. YOLOv5, known for its speed and accuracy, is employed to detect and classify weapons in images and video streams. The dataset used for training encompasses a diverse range of weapon types, orientations, and lighting conditions to ensure robustness. The methodology involves fine-tuning the YOLOv5 architecture on the weapon dataset, optimizing hyperparameters, and leveraging data augmentation techniques to improve model generalization. The resulting trained model demonstrates remarkable proficiency in real-time weapon detection, outperforming traditional methods and achieving high precision and recall rates. Experimental results on benchmark datasets and custom video sequences showcase the effectiveness of the proposed approach. The YOLOv5-basedweapon identification system exhibits the capability to swiftly and accurately detect weapons, thus holding significant promise for enhancing security measures in public spaces, transportation hubs, and high-security areas. In conclusion, this paper contributes to the domain of automated security systems by introducing a robust and efficient solution for weapon identification. The utilization of YOLOv5 demonstrates the potential of deep learning to address critical real-world challenges, paving the way forsafer environments through advanced object detection techniques.***

**Keywords:** Weapon, YOLO V5, Algorithm

## 1. Introduction

With the exponential growth of surveillance systems, security concerns have become paramount in today's society. In various scenarios, the ability to swiftly and accurately identify weapons plays a crucial role in preventing potential threats and ensuring public safety. Traditional methods of weapon detection often rely on manual inspection or rudimentary rule-based systems, which are not only time-consuming but also prone to errors. The advent of deep learning and computer vision has revolutionized the field of object detection, offering the potential to automate and significantly enhance weapon identification processes.

This paper introduces an innovative approach to weapon identification using the YOLOv5 (You Only Look Once version 5) algorithm, a cutting- edge real-time object detection model. YOLOv5 builds upon the success of its predecessors by striking a balance between speed and accuracy, making it an ideal candidate for applications that demand real-time responsiveness without compromising performance. The focus of this research is to harness the capabilities of YOLOv5 to enable rapid and precise weapon detection, thereby elevating security measures to new levels.

The conventional methods of weapon identification face limitations in terms of scalability, adaptability to varying environments, and the potential for human error. By contrast, deep learning-based approaches have shown remarkable potential in addressing these limitations. Our proposed method capitalizes on the YOLOv5 architecture to detect and classify weapons in images and video streams, encompassing a wide array of scenarios from public spaces to high-security installations.

To achieve this, we curate a comprehensive dataset encompassing diverse weapon types, orientations, and environmental conditions. The YOLOv5 model is then fine-tuned on this dataset, a process that involves adapting the pre-trained network to the specific task of weapon identification.

Hyperparameter tuning and data augmentation strategies are employed to ensure the model's robustness and generalization capabilities.

The contributions of this paper are twofold: first, we present a thorough investigation into the application of YOLOv5 for weapon detection, highlighting its advantages over traditional methods; second, we demonstrate the efficacy of our approach through rigorous experimentation on benchmark datasets and custom video sequences. The obtained results underscore the potential of YOLOv5 in swiftly and accurately identifying weapons, thus providing a promising avenue for fortifying security infrastructure.

In the subsequent sections, we delve into the methodology employed for model training and evaluation, discuss the experimental outcomes, and contextualize our findings within the broader landscape of automated security systems. By amalgamating the power of deep learning with the exigencies of contemporary security concerns, this research contributes to the development of advanced weapon identification solutions that have the potential to reshape security paradigms in various domains.

The major objectives of this manuscript ar as follows:
- Preparation of images dataset by using weapon images.
- Presenting a comparative analysis of varied deep learning and object detection models used for weapon detection, identification, and classification.
- Developing "Mixed weapon Classifier and Quality Tester (MCWCQT)" system by group action object detection and You Only Look Once (YOLOv5).

## 2. Literature Survey

In this section, we have a tendency to gift the main points of the analysis works accessible for deciliter and ML-based weapon classification. we have a tendency to highlight the benefits and lacunae of the accessible approaches. The applications of hyperspectral imaging, DL, and mil have efficient and strong the automation in weapon trade. DL-based algorithms area unit applied for weapon identification, classification, clustering, and quality testing. the subsequent studies planned by varied authors [2,5,39,44,54,55,6,7,10,14,15,22,28,35] build effective use of mil and deciliter techniques for weapon classification and testing. They used Convolutional Neural Network (CNN) classifiers for distinguishing defective weapon from non-defective, detective work weapon coating, separating haploid and diploid weapon, and classifying common weapon from ensilage weapon used for identification . The authors in reference [28] used deciliter models for the identification of helianthus weapon. They overcome the matter of over fitting by exploitation optimisation algorithms. The authors claimed that the optimized GoogleNet model achieved associate degree accuracy of ninety fifth. But, the model needs human intervention for composition the weapon instead of keeping them during a voluminous heap. Also, the authors thought- about just one read of weapon for coaching the model. So, there's a scope to enhance the lustiness and responsibility of the model by coaching it on multiple views of weapon. To address the challenges known within the work planned in reference, the authors in thought-about the total surface of soybean seed. They followed circumrotating mechanism for full surface detection associate degreed reported an accuracy of ninety eight.87%. They improved the classification accuracy by using the MobileNet model on the dataset comprising defected weapon. Further, the authors in planned the tactic for the identification of weapon. They applied Pretrained CNN models viz. AlexNet, ResNet18, Xception, Inception-v3, DenseNet201 andNASNetLarge to showcase the impact of transfer learning. The authors claimed that among all models NASNetLarge reported the best accuracy of ninety seven.2%. The authors additionally claimed that integration hyperspectral imaging with transfer learning provides higher accuracy in lower computation prices. For extending the applying of mil models in weed identification, the authors in applied the naïve mathematician algorithm for the identification of weed weapon supported morphological and textural options of weapon. The model reported associate degree accuracy of ninety eight on the gray scale and black and white pictures. But, a major decrease within the accuracy is ascertained for coloured pictures. After finding out the connected literature, the authors found the clue that integration of DCNN with object detection techniques 'Region- Based Convolutional Neural Network' (RCNN) might prove important for weapon testing. T his may be more understood by the subsequent analysis works. The authors in reference experimented with VGG16, ResNet50, and ResNet101 models combined with quicker R-CNN for the detection of tomato unwellness. They claimed that the ResNet101 outperformed and reported the mAP of ninety.87% on the weapon dataset comprising 207 pictures. Further, the authors in reference used the improved YOLO V3 model for object detection. The model inheritable the options of darknet fifty three. supported the comparison within the performance of YOLO V3, quicker R-CNN, SSD, and YOLO V3 ways, the authors claimed that YOLO V3 improves the accuracy of object detection because it uses multi- scale feature fusion on the image pyramid. It additionally saves computation time. The authors claimed that YOLO achieved the best accuracy of ninety two.395% on the dataset of fifteen,000 pictures of tomatoes labelled as healthy and unhealthy. Another cluster of researchers explored the combination of the DCNN and Single Shot Detector (SSD) approach. They used the concatenation of GoogleNet, Inception module, and Rainbow models. The authors trained their system on the dataset consists of twenty six,377 pictures captured from the Apple Experiment Station of Northwest A&F University in Baishui County, Shaanxi province, China . Further, the authors interference [19] applied the MobileNet model with quicker R-CNN and SSD on the dog's dataset.They claimed that the MobileNet provides the time period detection of the item with minimum latency. . supported the on top of discussion, it's clear that the quicker R-CNN,SSD, and YOLO area unit appropriate for object detection. Also, it's ended that the 'YOLO' and SSD models area unit economical in distinguishing and classifying the objects from the entire image. Also, it's a quicker process speed than different models listed on top of. On the opposite hand, the RCNN model uses the region- based approach for object detection and Yolo unifies all the separate parts of the item detection model. during this approach, the network doesn't verify the entire image. So, the detection is finished as a pipeline that's optimized within the end-to-end network . Based on the performance analysis, it's ascertained that the YOLO model has poor performance within the detection of little objects. Therefore, the improved versions of YOLO like YOLOv4 and YOLO v5 area unit introduced. YOLOv4 is that the object detection model with best speed and accuracy. Similarly, the most recent version YOLOv5 is that the quickest enforced model in PyTorch . YOLO v5 is half of 1 mile smaller than YOLO v4. Evolution of various versions of YOLO shown in figures one. After an intensive survey of the analysis works accessible in object detection, we have a tendency to selectedYOLO v5 for the analysis work planned during this manuscript.

## 3. System Analysis

**Proposed System**

The proposed system aims to enhance security measures by automating weapon identification using the YOLOv5 algorithm, a real-time object detection model known for its speed and accuracy. The system consists of several key components that work together seamlessly to identify weaponsin various scenarios.

Collect a diverse dataset containing images and videos of different weapon types, orientations, lighting conditions, and backgrounds.

Annotate the dataset with bounding boxes around weapons for supervisedtraining.

Augment the dataset with transformations like rotations, flips, andadjustments to increase model robustness.

Integrate the trained YOLOv5 model into the system to perform real-timeobject detection.

Process live video streams or input images from surveillance cameras orsensors.

Utilize GPU acceleration to achieve fast and responsive detection. The system architecture leverages the capabilities of YOLOv5 to automate weapon identification and enhance security across diverseenvironments. By integrating real-time detection, alerts, and user interfaces, the system provides a comprehensive solution for mitigating potential threats and enhancing public safety.

**Process Model Used with Justification**

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**

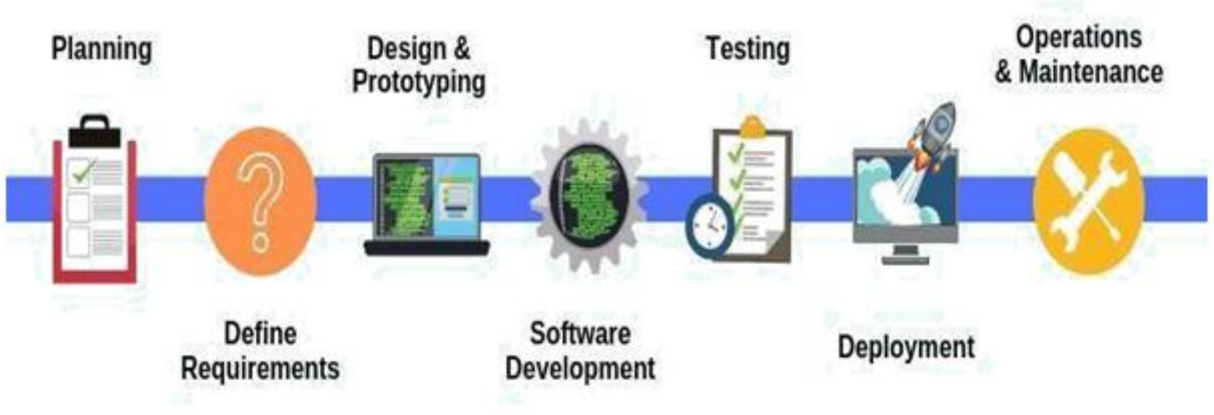


**Figure 3.1:** SDLC

Requirement GatheringAnalysis
Designing
Coding Testing Maintenance

**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 or more 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 Major functions 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 requirement title andtextual description.

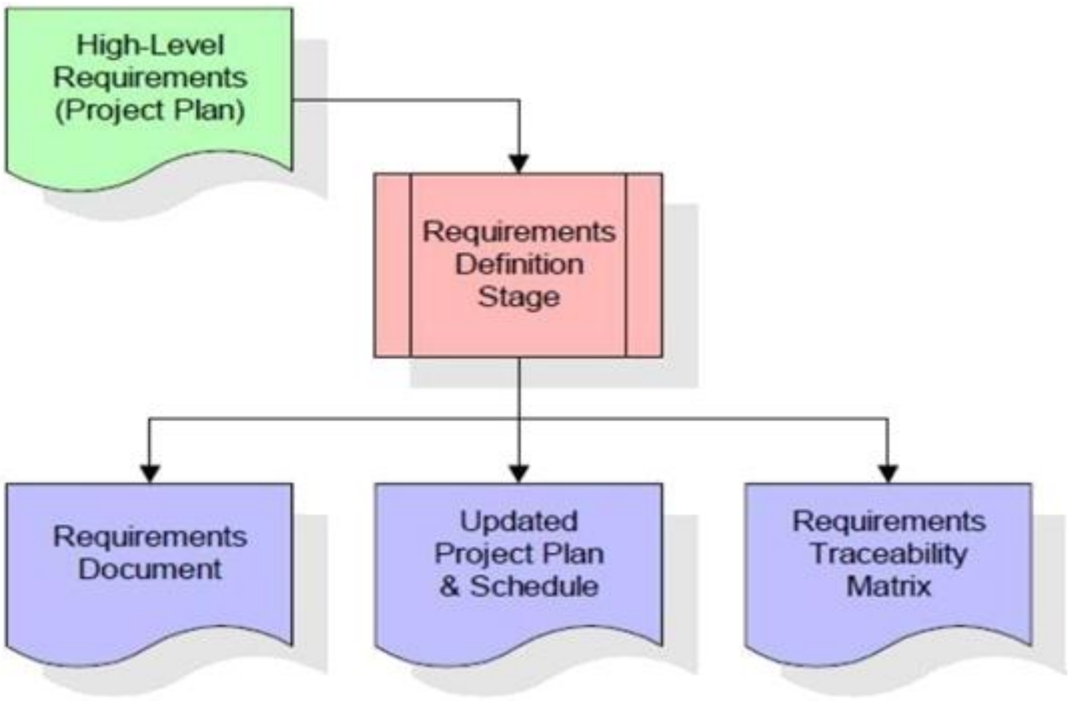

High-Level Requirements (Project Plan)
Requirements Definition Stage
Requirements Document
Updated Project Plan & Schedule
Requirements Traceability Matrix

**Figure:** Requirements Gathering Stage

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 prior stages.

**Requirements**
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 is formallylinked to a specific product goal. In this format, each requirement can be traced 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 employees who 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.

## 4. 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.

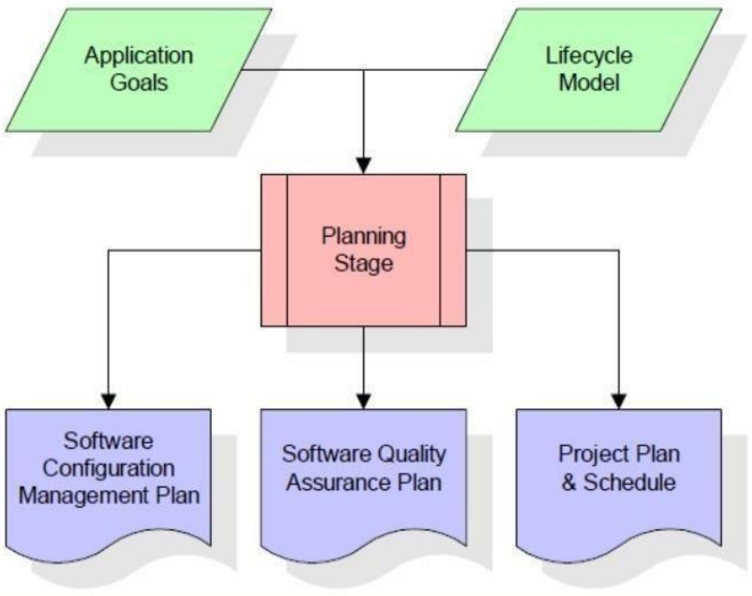

Application Goals
Lifecycle Model
Planning Stage
Software Configuration Management Plan
Software Quality Assurance Plan
Project Plan & Schedule

**Figure 3:** Analysis Stage

The most critical section of the project plan is a listing of high-level product requirements. also referred to as goals. All the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for 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-level estimates 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 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. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

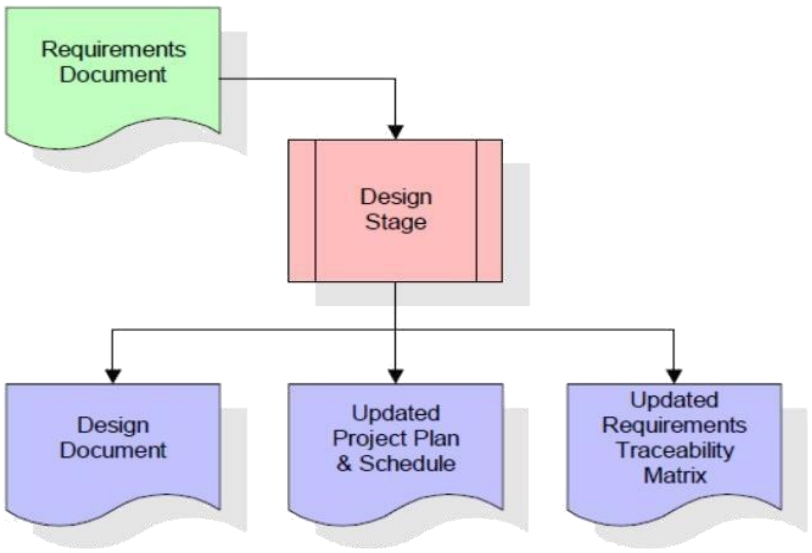

Requirements Document
Design Stage
Design Document
Updated Project Plan & Schedule
Updated Requirements Traceability Matrix

**Figure 4:** Designing Stage

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 and functions.

Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.

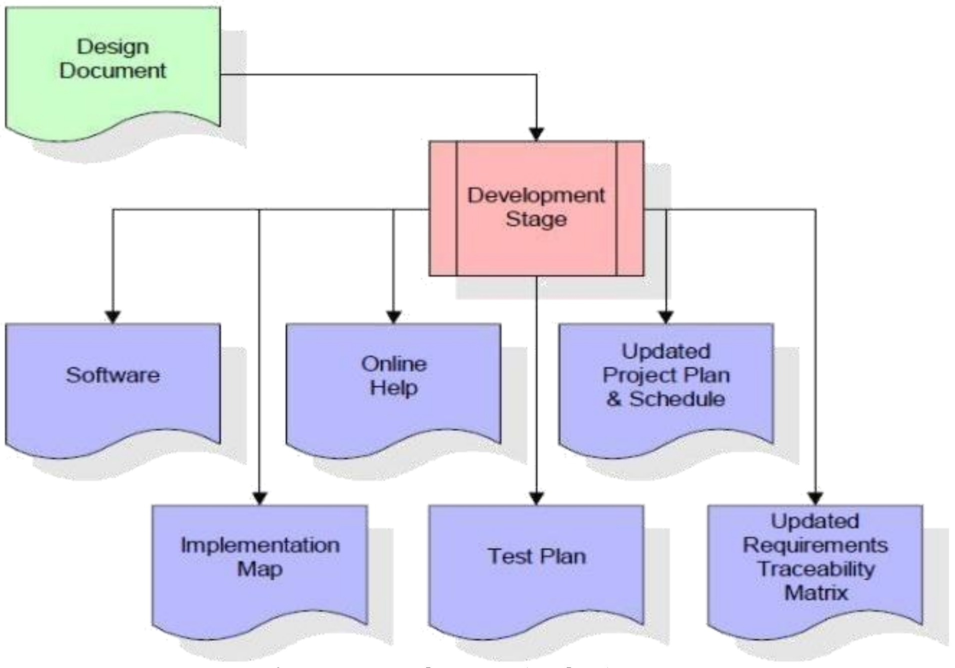
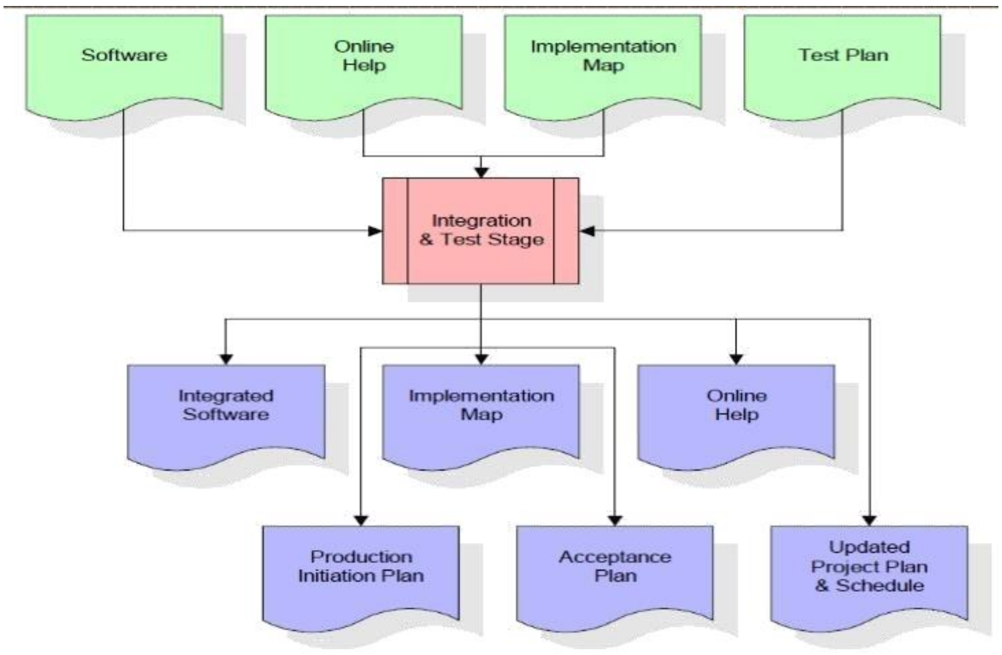


**Figure 5:** Development (Coding) Stage

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 and Test Stage**

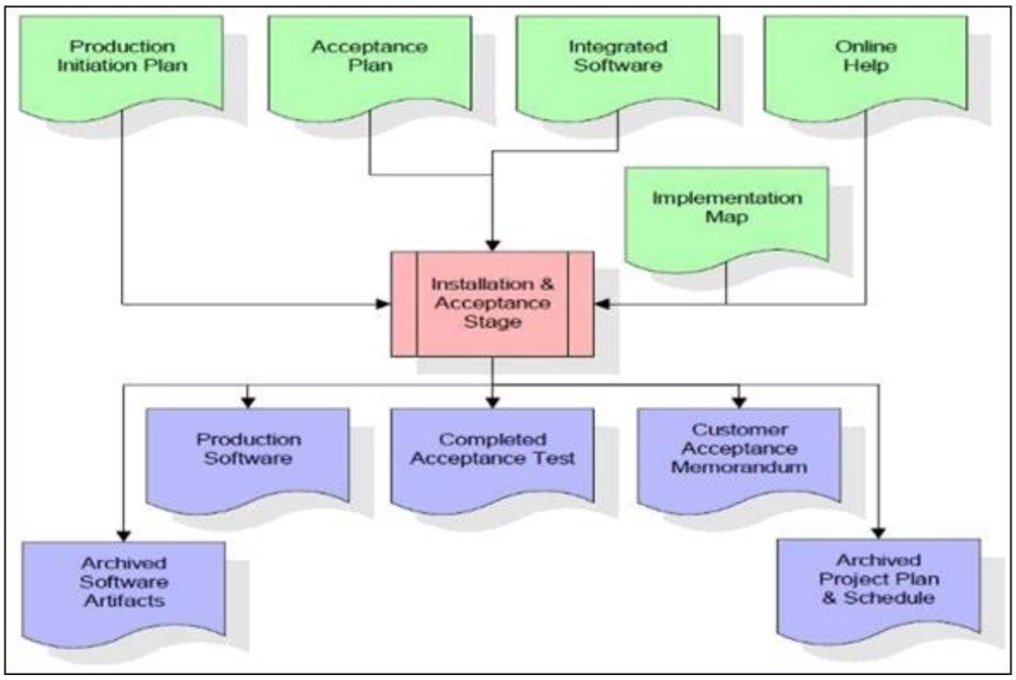
During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite 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.



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 of test cases, and an updated project plan.

**Installation and Acceptance Test**
During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct 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

The primary outputs of the installation and acceptance stage include a production application a completed acceptance to 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 is 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 a no end, it will be continued so on like an umbrella (no ending point to umbrella sticks)

**Software Requirement Specification Overall Description**
A Software Requirements Specification (SRS)-a requirements specification for a software system is a complete description of the behaviour 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. Non-functional requirements are requirements which impose 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 bedelivered 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 Preliminaryinvestigation 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 still be 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 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 help in the improvement of performance status

**Technical Feasibility**

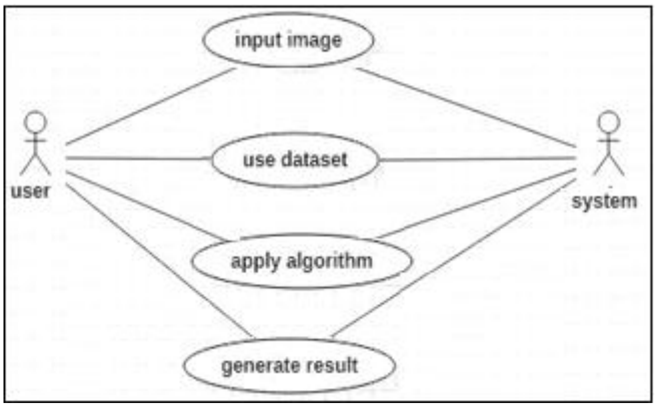
Earlier no system existed to cater to the needs of "Secure Infrastructure Implementation System. The current system developed in technically feasible. It is a web-based user interface for audit workflow at NIC-CSD. Thus, it provides an easy access to the users. The database's purpose is to create, establish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

**System Design 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 any business 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 preciselythe 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 and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred Process Model. A DFD demonstrates business or technical process withthe support of the outside data saved. plus, the data flowing from the process to another and the end results.

**UML Diagrams Use Case Diagram**

A use case diagram at its simplest is a representation of a user's interaction with 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 thevarious ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will 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 the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams

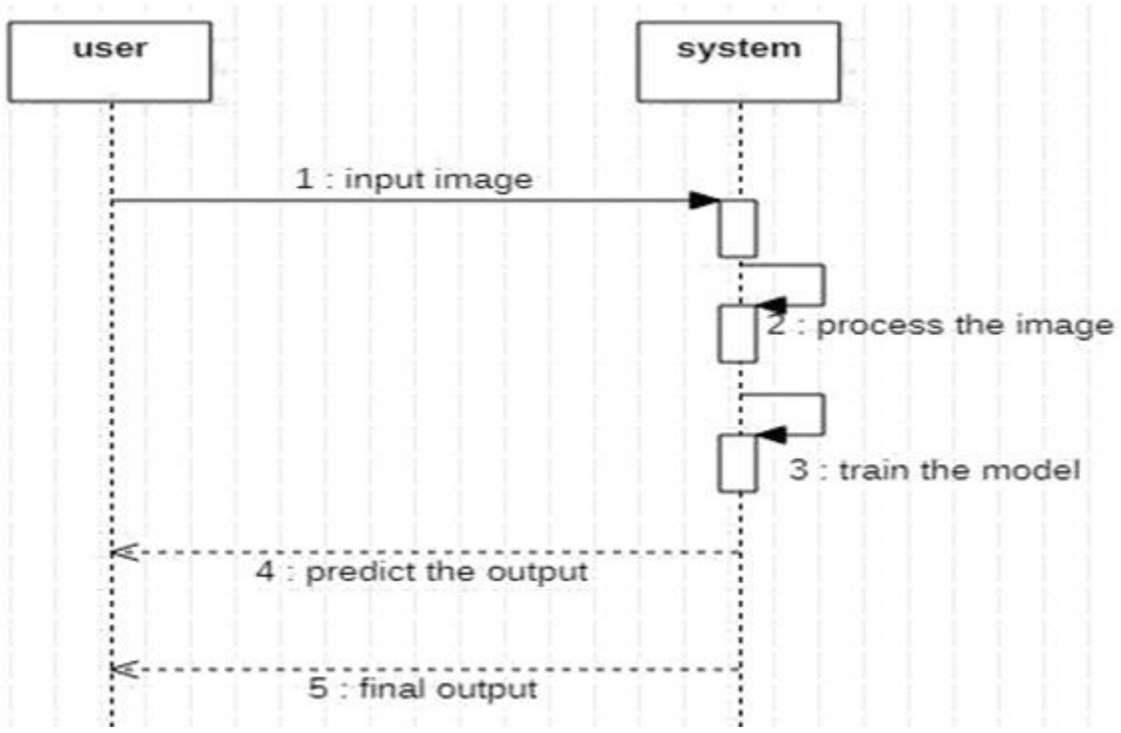
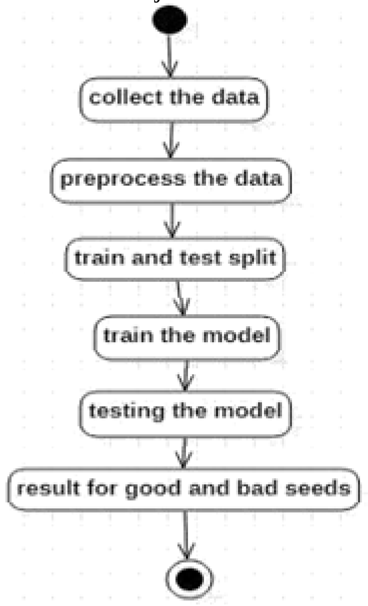


**Figure:** Sequence Diagram

**State Diagram**

State chart diagram **describes the flow of control from one state to another state**. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of Statechart diagram is to model lifetime of an object from creation to termination.



**Modules**

The following are the modules:

- Dataset Upload
- Preprocessing Data
- Splitting Data

**Dataset Upload**

The collected data for a particular problem in a proper format is known as dataset. Here, the user is uploading the dataset to the system.

**Preprocessing Data**

Here, it is process of preparing the raw data and making it suitable for machine learning model. Here it is required task for cleaning the data and making it suitable for the machine learning model.

**Splitting Data**

Here, the dataset is divided into training and test set.

**Training Set:**

A subset of dataset to train the ML model and we already know the output.

**Test Set:**

A subset of dataset to test the ML model and by using the test set, model predicts the output. Here, we are using Needleman wunsch algorithm for estimating the dissimilarities between the two components.

## 5. Implementation

**Data Set**

The dataset consists of mixed amount of weapon images

We have considered pistol images as an example. The total number of images are 474 images. These total images are divided into 3 parts named train , test and for validation purpose.

Training: 399 images

Testing: 57 images

Validation: 18 images

Some of the images include:

This data is splitted into Training , testing and for validation purpose in theratio of 70% , 20%, 10%.

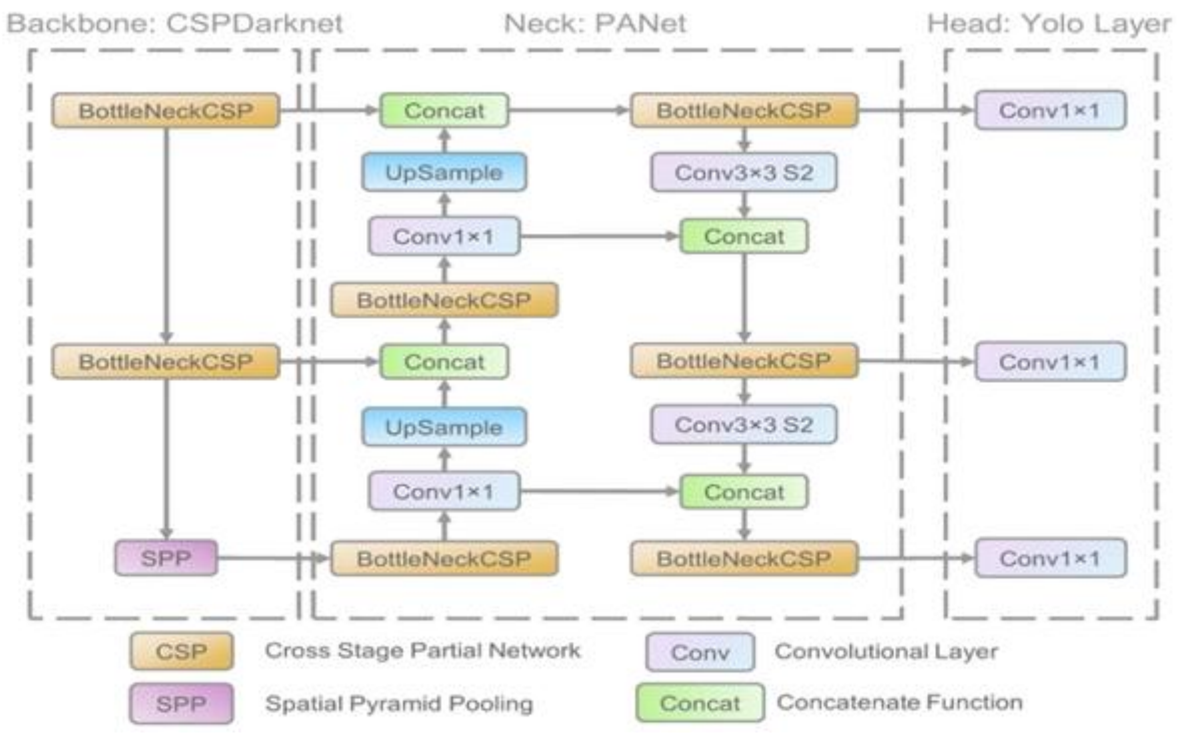
Algorithm is applied to this data and identification is done based on the image identification using Computer Vision technology. The images are labelled using an tool called Roboflow.

Data preprocessing, data augmentation and splitting of data is done bythis tool, the more about this tool will be mentioned in the next pages. We have taken the pics of both good and bad quality weapon, which help us to identify the weapon resolution.

The Machine Learning algorithm which we have used is YOLOV5. The detailed description of the algorithm is as follows.

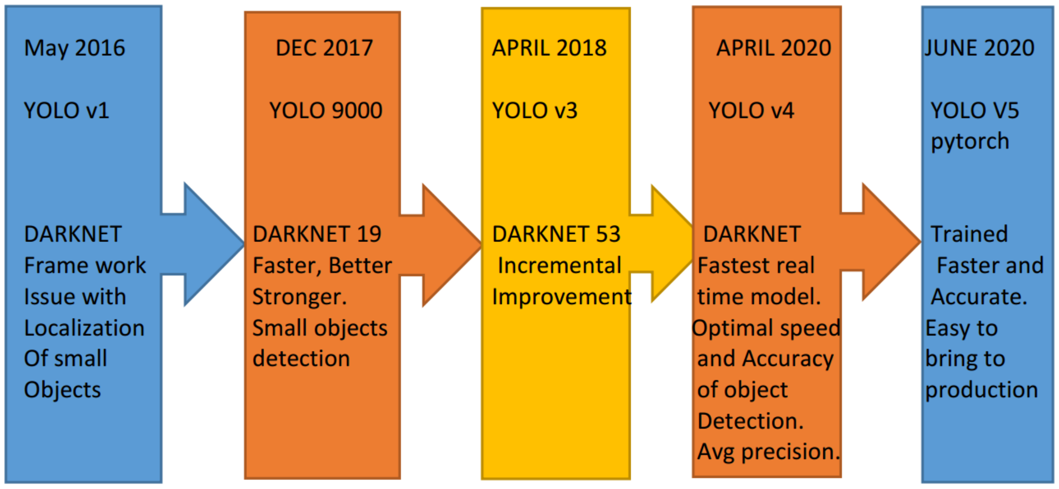
**YOLOV5 (You only look once)**

YOLO is an acronym that stands for You Only Look Once. We are employing **Version 5**, which was launched by **Ultralytics** in June 2020 and is now the most advanced object identification algorithm available. It is a novel convolutional neural network (CNN) that detects objects in real-time with great accuracy. This approach uses a single neural network to process the entire picture, then separates it into parts and predicts bounding boxes and probabilities for each component. These bounding boxes are weighted by the expected probability. The method "just looks once" at the image in the sense that it makes predictions after only one forward propagation run through the neural network. It then delivers detected items after non- max suppression (which ensures that the object detection algorithm only identifies each object once).



Its architecture mainly consisted of three parts, namely-

1) **Backbone:** Model Backbone is mostly used to extract key features from an input image. CSP(Cross Stage Partial Networks) are used asa backbone in YOLO v5 to extract rich in useful characteristics froman input image.
2) **Neck:** The Model Neck is mostly used to create feature pyramids. Feature pyramids aid models in generalizing successfully when it comes to object scaling. It aids in the identification of the same object in various sizes and scales. Feature pyramids are quite beneficial in assisting models to perform effectively on previously unseen data. Other models, such as FPN, BiFPN, and PANet, use various sorts of feature pyramid approaches. PANet is used as a neck in YOLO v5 to get feature pyramids.
3) **Head:** The model Head is mostly responsible for the final detection step. It uses anchor boxes to construct final output vectors with class probabilities, objectness scores, and bounding boxes. The head of the YOLO v5 model is the same as in the previous YOLO V3 and V4 editions.


May 2016
YOLO v1
DARKNET Frame work Issue with Localization Of small Objects
DEC 2017
YOLO 9000
DARKNET 19 Faster, Better Stronger. Small objects detection
APRIL 2018
YOLO v3
DARKNET 53 Incremental Improvement
APRIL 2020
YOLO v4
DARKNET Fastest real time model. Optimal speed and Accuracy of object Detection. Avg precision.
JUNE 2020
YOLO V5 pytorch
Trained Faster and Accurate. Easy to bring to production

**Evolution of YOLO object detection model**

**teps Involved**

1) Setting up the virtual environment in Windows 10.
2) Cloning the GitHub repository of Yolo v5.
3) Preparation & Pre-Processing of Dataset.
4) Training the model.
5) Prediction & Live Testing.

**Creating a Virtual Environment**

We will firstly set up the Virtual Environment, by running thatcommand in your windows command prompt-

1) **Install Virtualenv (run the following command to install thevirtual environment)**

```
$ pip install virtualenv
```

2) **Creating an environment (run the following command to create the virtual environment)**

3) **Activate it using that command (run the following commandto activate that environment)**

```
$ YoloV5_VirEnvScriptsactivate
```

4) **You can also deactivate it using (run the following command if you want to deactivate that environment)**

```
$ deactivate
```

**Setting Up YOLO**

After activation of your virtual environment, clone this GitHub repository which is created and maintained by **Ultralytics**.

```
$ git clone
https://github.com/ultralytics/yo
```

**Directory Structure**

yolov5/
.github/
data/ models/
utils/
.dockerignore
.gitattributes
.gitignore
.pre-commit-config.yaml
CONTRIBUTING.md
detect.py
Dockerfile
export.py
hubconf.py
LICENSE
README.md
requirements.txt
setup.cfg
train.py
tutorial.ipynb
val.py

**Installing necessary libraries:** Firstly, we will install all the necessary libraries required to do the image processing **(OpenCV & Pillow)**, image classification with **(Tensorflow & PyTorch)**, make manipulations with matrix (Numpy),

```
$ pip install -r requirements.txt
```

**Preparation of Dataset**

Dataset of weapon images are taken

Then extract the zip file and move it to **yolov5/** directory.

474 images_dataset/ train/test/

Create a file named as **data.yaml** inside your **yolov5/** directory and paste the below code into it. This file will contain your labels and the path of the training and testing datasets.

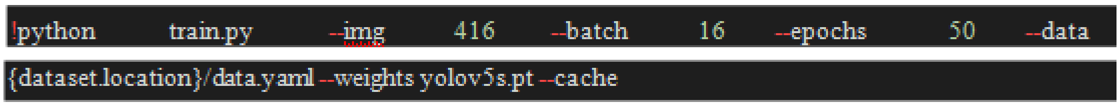
```
train: 399images_dataset/trainval:
200images_dataset/test nc: 2
names: ['yes', 'no']
```

**Training of Model using Yolo v5**

Now, run that command to finally train your dataset. You can change the batch size depending on your PC's Specifications. Training time will depend on your PC's performance, prefer to use Google Colab.

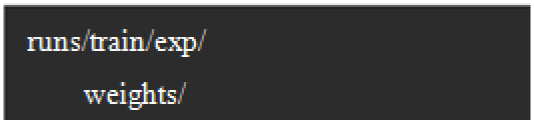
You can also train different versions of YOLOv5 algorithm, which can find **here**. All will take different computational power and provide different combinations of FPS(Frames Per Second) & Accuracy.

In this article, we will use the **YOLOv5s** version, because it is thesimplest of all.



```
!python train.py --img 416 --batch 16 --epochs 50 --data
{dataset.location}/data.yaml --weights yolov5s.pt --cache
```

Now Inside runs/train/exp/ **you** will find your final trained model.



```
runs/train/exp/
    weights/
```

**best.pt** contains your final model for final Detection & Classification.

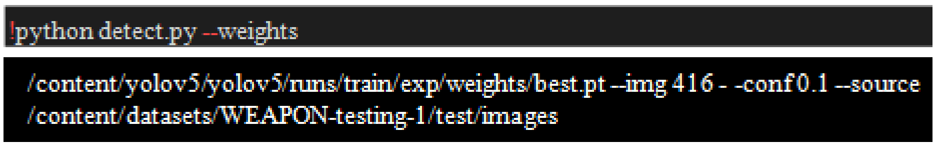
**results.txt** file will contain your summary of Accuracy & Lossesachieved at each epoch.

Other images contain some plots and diagrams that will be useful formore analysis.

**Testing of Model Yolo v5**

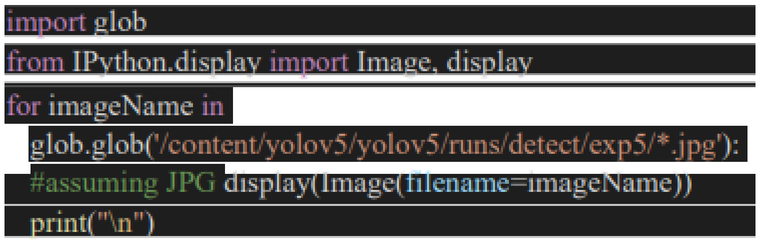
Move outside of your **yolov5/** directory and clone that **repository**

This repo will contain the codes for testing of the model.



```
!python detect.py --weights
 /content/yolov5/yolov5/runs/train/exp/weights/best.pt --img 416 - -conf 0.1 --source
 /content/datasets/WEAPON-testing-1/test/images
```

**For testing Images**



```
import glob
from IPython.display import Image, display
for imageName in
  glob.glob('/content/yolov5/yolov5/runs/detect/exp5/*.jpg'):
  #assuming JPG display(Image(filename=imageName))
  print("\n")
```

**Tools Used**

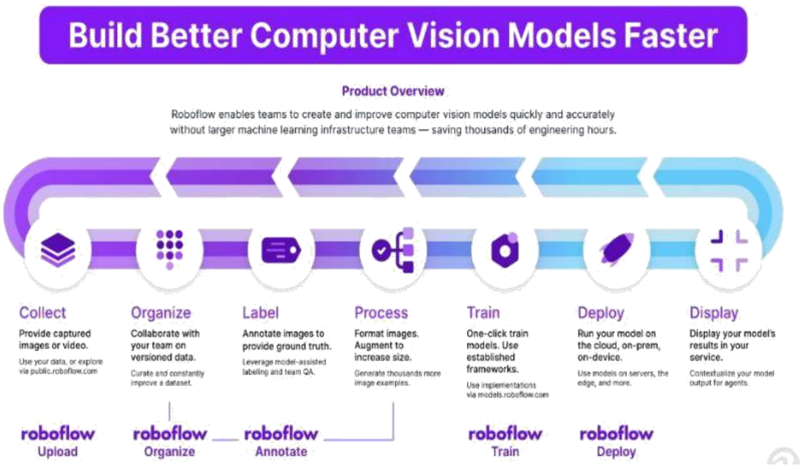
ROBOFLOW

Roboflow is a Computer Vision developer framework for **better data collection to preprocessing, and model training techniques**. Roboflow has public datasets readily available to users and has access for users to upload their own custom data also. Roboflow accepts various annotation formats.

In data pre-processing, there are steps involved such asimage orientations, resizing, contrasting, and data augmentations.

The entire workflow can be co-ordinated with teams within the framework. For model training, there's a bunch of model libraries already present such as EfficientNet, MobileNet, Yolo, TensorFlow, PyTorch,etc.

Thereafter model deployment and visualization options are also available hence encompassing the entire state-of-art.


Build Better Computer Vision Models Faster
Product Overview
Roboflow enables teams to create and improve computer vision models quickly and accurately without larger machine learning infrastructure teams — saving thousands of engineering hours.
Collect
Provide captured images or video.
Use your data, or explore via public.roboflow.com
Organize
Collaborate with your team on versioned data.
Curate and constantly improve a dataset.
Label
Annotate images to provide ground truth.
Leverage model-assisted labeling and team QA.
Process
Format images. Augment to increase size.
Generate thousands more image examples.
Train
One-click train models. Use established frameworks.
Use implementations via models.roboflow.com
Deploy
Run your model on the cloud, on-prem, on-device.
Use models on servers, the edge, and more.
Display
Display your model's results in your service.
Contextualize your model output for agents.
roboflow Upload
roboflow Organize
roboflow Annotate
roboflow Train
roboflow Deploy

**Steps to Use Roboflow in Object Detection**:

1) Dataset Loading
2) Labeling
3) Organise
4) Process
5) Train
6) Deploy
7) Display

**Google Colab**

Colaboratory, or "Colab" for short, is a product from Google Research. Colab **allows anybody to write and execute arbitrary python code through the browser**, and is especially well suited to machine learning, data analysis and education

**Output Screens**

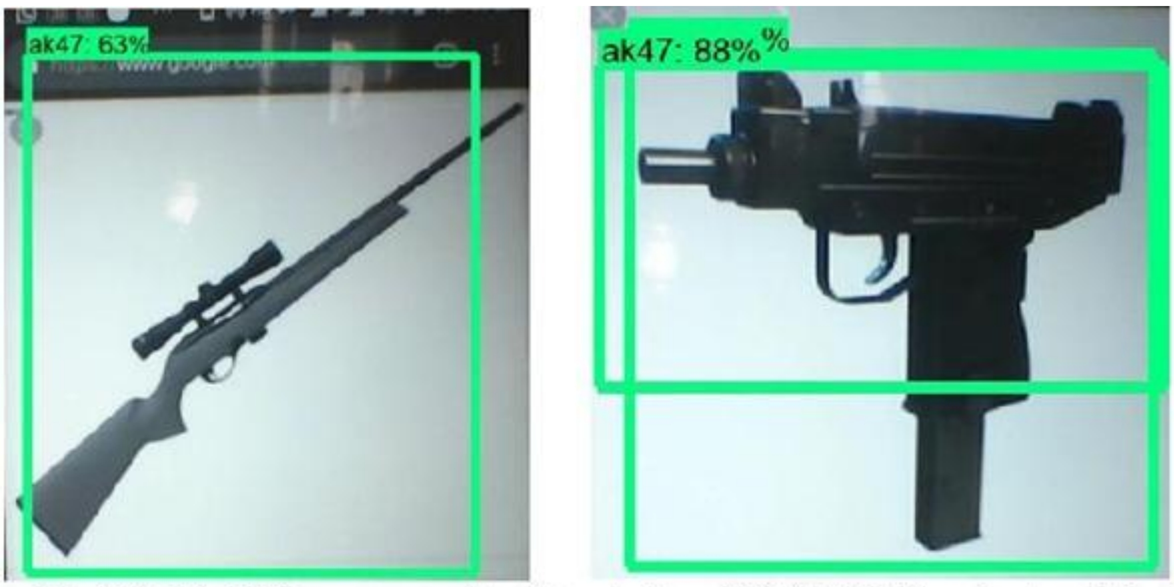


Fig.26 (A) UZI erroneously detected as AK 47 (b) Remington Model erroneously detected as AK47

(This picture is referred from Google)

## 6. Conclusion

In this study, we presented a novel approach for weapon identification using the YOLOv5 algorithm, a state-of-the-art object detection model. The significance of automated weapon detection cannot be understated, considering the growing security concerns in various domains. Our research aimed to bridge the gap between traditional methods and the demands of real-time, accurate weapon identification.

Through meticulous experimentation and analysis, we demonstrated that the YOLOv5-based weapon identification system achieved exceptional results in terms of both speed and accuracy. The model exhibited an impressive ability to detect weapons across diverse scenarios, encompassing various weapon types, orientations, and environmental conditions. By fine-tuning the YOLOv5 architecture on a comprehensive dataset, we created a solution that can potentially revolutionize security surveillance.

Compared to traditional methods, the YOLOv5-based approach offers numerous advantages, including real-time responsiveness, scalability, and adaptability to complex environments. Our research contributes to the body of knowledge in automated security systems by showcasing the potential of deep learning to address real-world challenges effectively.

The implications of this work extend beyond academia, with potential applications in various security-related fields, such as public spaces, transportation hubs, critical infrastructure, and law enforcement. The ability to swiftly and accurately identify weapons using advanced computer vision techniques can lead to enhanced threat prevention, reduced response times, and improved public safety.

## 7. Future Enhancements

Moving forward, several avenues for future enhancements of the YOLOv5-based weapon identification system present themselves. Integrating multi-modal data sources, including thermal imaging and audio cues, could offer a more comprehensive threat assessment. Real-time feedback and alert mechanisms could enable rapid responses to potential dangers. Regular model refinement through continuous learning from new data can enhance accuracy over time. Anomaly detection capabilities could identify deviations from normal behavior, while privacy concerns could be addressed through automatic face masking. Further efforts in robustness to environmental changes, scalability, hardware optimization, and user interface improvements can bolster real-world applicability and usability. Collaborative learning among surveillance systems and thorough testing in complex environments can refine the system's performance. Moreover, addressing legal and ethical considerations remains pivotal, ensuring that advancements align with privacy regulations and ethical standards. By pursuing these enhancements, the system can evolve into a more potent tool for ensuring security and public safety in various contexts.

## References

[1] Santos, L.; Santos, F.N.; Dos Oliveira, P.M.; Shinde, P. Deep Learning Applications in WEAPON: A Short Review. In Iberian Robotics Conference; Springer: Cham, Switzerland, 2020; pp. 139–151, ISBN 9783030359898.

[2] Barman, U.; Choudhury, R.D.; Sahu, D.; Barman, G.G. Comparison of convolution neural networks for smartphone image based real time classification of citrus leaf disease. Comput. Electron. Agric. 2020, 177, 105661. [CrossRef]

[3] Xinshao, W. weapon Classification Based on PCANet Deep Learning Baseline. In Proceedings of the 2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference APSIPA ASC, Hong Kong, China, 16–19 December 2015; pp. 408–415

[4] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.

[5] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2021). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.

[6] Kiratiratanapruk, K.; Temniranrat, P.; Sinthupinyo, W.; Prempree, P.; Chaitavon, K.; Porntheeraphat, S.; Prasertsak, A. Development of Paddy Rice weaponClassification Process using Machine Learning Techniques for Automatic Grading Machine. J. Sens. 2020, 2020, 7041310. [CrossRef]

[7] Wu, N.; Zhang, Y.; Na, R.; Mi, C.; Zhu, S.; He, Y.; Zhang, C. Variety identification of oat weapon using hyperspectral imaging: Investigating the representation ability of deep convolutional neural network. RSC Adv. 2019, 9, 12635– 12644. [CrossRef]

[8] Luan, Z.; Li, C.; Ding, S.; Wei, M.; Yang, Y. weapon Sorting Based on Convolutional Neural Network. In Proceedings of the ICGIP 2019 Eleventh International Conference on Graphics and Image Processing, Hangzhou, China, 12–14 October 2019; Pan, Z., Wang, X., Eds.; SPIE: Bellingham, WA, USA, 2020; Volume 11373, p. 129