**Smart surveillance Violence Detection using alert system**

**Abstract**

The rapid advancement of computer vision technologies has enabled automated solutions for critical tasks such as Violence detection in surveillance systems. This project focuses on implementing a Violence detection system using the YOLOv8 (You Only Look Once, Version 8) model trained on the Roboflow Violence detection object dataset. The system aims to detect and classify Violences in real-time video feeds or images with high accuracy and efficiency.

**Problem Statement:**

Effectively identifying and reducing possible risks posed by concealed Violences is a major concern facing public safety today. The ability of traditional security screening techniques to quickly and precisely detect hidden Violences in real-time situations is frequently compromised. This discrepancy necessitates the development of an advanced artificial intelligence and deep learning Violence detection system. Because the current systems are not sophisticated enough to offer effective and trustworthy threat detection, there may be weaknesses in different security configurations. By creating a reliable and moral Violence detection system A device capable of precisely identifying Violencery and monitoring them in real-time, our project seeks to close this crucial gap in public safety protocols and reduce the risks that come with bringing a hidden Violence in an array of contexts.

**Proposed System:**

Precise Violence detection and categorization are the main goals of the proposed implementation. Another reason accuracy matters is because a false alarm might have unintended consequences. Selecting the appropriate strategy is necessary to balance speed and precision appropriately. From the input video, frames are taken off. Before an object is detected, the frame differencing procedure creates the bounding box. The YOLOv8 algorithm achieved unprecedented levels of accuracy and performance detection.

**Key Objectives:**

* **Real-Time Violence Detection:** Utilize YOLOv8's state-of-the-art object detection capabilities to identify Violences in various scenarios, including crowded or complex environments.
* **Dataset Utilization:** Leverage the pre-labeled and curated Violence detection dataset from Roboflow to train and fine-tune the YOLOv8 model.
* **Enhanced Safety and Security:** Provide an automated tool that aids security personnel in identifying potential threats, minimizing response time, and enhancing situational awareness.

**Methodology:**

* **Dataset Preparation:** The Roboflow Violence detection dataset will be pre-processed and augmented to improve model robustness against diverse lighting, angles, and occlusions.
* **Model Selection and Training:** YOLOv8's lightweight architecture will be fine-tuned on the Violence dataset to ensure accurate detection while maintaining real-time performance.
* **Evaluation:** The model will be evaluated using standard metrics such as mAP (mean Average Precision), precision, and recall on test datasets.
* **Deployment:** The trained model will be integrated into a system capable of processing live video streams or pre-recorded footage.

**System Configuration:**

**Hardware Requirements:**

Processor :           Any Update Processor

Ram :           Min 8 GB

Hard Disk :           Min 100 GB

**Software Requirements:**

Operating System       :         Windows family

Technology                 :           Python 3.8

Front-end Technology : HTML, CSS, JS

IDE : PyCharm

Web framework : Flask

**INTRODUCTION**

In recent years, public safety has become a significant concern due to the increasing instances of violence in various settings such as streets, schools, offices, and public gatherings. Traditional surveillance systems rely on human monitoring, which is prone to fatigue and inefficiency. To address this challenge, Smart Surveillance for Violence Detection using an Alert System integrates Deep Learning with computer vision to create an automated and intelligent monitoring system.

This project leverages YOLOv8 (You Only Look Once), a state-of-the-art object detection algorithm, to accurately detect violent activities in real time. The system is developed using Python 3.8, with Flask as the web framework for backend processing and MySQL for database management. The front-end interface is designed using HTML, CSS, and JavaScript, ensuring an interactive and user-friendly experience.

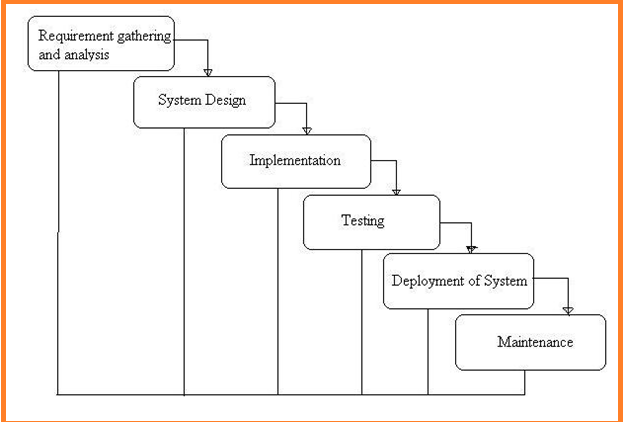
Upon detecting violent behavior, the system triggers an instant alert mechanism, notifying the concerned authorities through automated messages or emails. This enhances security by allowing prompt action, thereby preventing potential threats and ensuring public safety.

By integrating machine learning, deep learning, and web technologies, this project provides an efficient, scalable, and reliable solution for modern surveillance systems. It is particularly useful for applications in smart cities, corporate offices, educational institutions, and public spaces, reinforcing safety and reducing crime rates through proactive monitoring.

**LITERATURE SURVEY**

Due to the quick development of deep learning and artificial intelligence (AI) technologies, the field of smart surveillance has seen tremendous growth in recent years. Conventional surveillance systems mostly depend on human operators to keep an eye on video feeds for any threats or illegal activity. Although human monitoring works well in controlled environments, it is naturally constrained by things like exhaustion, attention, and the inability to cover several regions at once. Large-scale surveillance networks, like those found in shopping centers, schools, public transportation systems, and metropolitan areas, where ongoing monitoring is essential to maintaining public safety, make these flaws much more noticeable. Machines can now "learn" from massive volumes of data and provide intricate predictions thanks to deep learning, which has completely changed the field of computer vision and pattern recognition. Deep learning methods, especially Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have demonstrated impressive results in the surveillance domain in terms of video stream analysis, object identification, and the detection of anomalous behaviors. RNNs are excellent at temporal analysis, detecting patterns of activities across time, but CNNs are especially well-suited for spatial analysis, such as identifying gestures or actions. An intelligent, automated system that can identify violence in real time is more important than ever because of the growing threat of violence in public areas. An important tool for public safety, such a system can increase the effectiveness of current surveillance infrastructure by decreasing the need for human monitoring and increasing the speed and precision of incident detection.

**SYSTEM ANALYSIS**



**What is Waterfall Model?**

Waterfall Model is a sequential model that divides software development into different phases. Each phase is designed for performing specific activity during SDLC phase. It was introduced in 1970 by Winston Royce.

**Requirements:**

The first phase involves understanding what needs to design and what is its function, purpose, etc. Here, the specifications of the input and output or the final product are studied and marked.

**System Design:**

The requirement specifications from the first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The software code to be written in the next stage is created now.

**Implementation:**

With inputs from system design, the system is first developed in small programs called units, which are integrated into the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

**Integration and Testing:**

All the units developed in the implementation phase are integrated into a system after testing of each unit. The software designed, needs to go through constant software testing to find out if there are any flaws or errors. Testing is done so that the client does not face any problem during the installation of the software.

**Deployment of System:**

Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.

**Maintenance:**

This step occurs after installation, and involves making modifications to the system or an individual component to alter attributes or improve performance. These modifications arise either due to change requests initiated by the customer, or defects uncovered during live use of the system. The client is provided with regular maintenance and support for the developed software.

**Feasibility study**

A feasibility analysis evaluates the project’s potential for success; therefore, perceived objectivity is an essential factor in the credibility of the study for potential investors and lending institutions.

**Technical Feasibility**

Technical resources need for project Development

* Windows family Operating System
* Python 3.6 Technology
* Vs Code
* Mysql
* Sqlyog

**Economic Fesibility**

Cost/ benefits analysis of the project as over project is academic project we will not have only basic cost for learning of the technologies

**Operational Feasibility**

This assessment involves undertaking a study to analyze and determine whether and how well the organization’s needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

**SOFTWARE OVER VIEW:**

**History of Python**

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

**Input as CSV File**

Reading data from CSV(comma separated values) is a fundamental necessity in Data Science. Often, we get data from various sources which can get exported to CSV format so that they can be used by other systems. The Panadas library provides features using which we can read the CSV file in full as well as in parts for only a selected group of columns and rows.

The CSV file is a text file in which the values in the columns are separated by a comma. Let's consider the following data present in the file named input.csv.You can create this file using windows notepad by copying and pasting this data. Save the file as input.csv using the save As All files(\*.\*) option in notepad.

import pandas as pd

data= pd.read\_csv('path/input.csv')

print(data)

**Operations using NumPy**

NumPy is a Python package which stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Using NumPy, a developer can perform the following operations −

* Mathematical and logical operations on arrays.
* Fourier transforms and routines for shape manipulation.
* Operations -related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

## Key Features of Pandas

* Fast and efficient DataFrame object with default and customized indexing.
* Tools for loading data into in-memory data objects from different file formats.
* Data alignment and integrated handling of missing data.
* Reshaping and pivoting of date sets.
* Label-based slicing, indexing and subsetting of large data sets.
* Columns from a data structure can be deleted or inserted.
* Group by data for aggregation and transformations.
* High performance merging and joining of data.
* Time Series functionality.

# **Python Flask Tutorial**



Flask Tutorial provides the basic and advanced concepts of the Python Flask framework. Our Flask tutorial is designed for beginners and professionals.

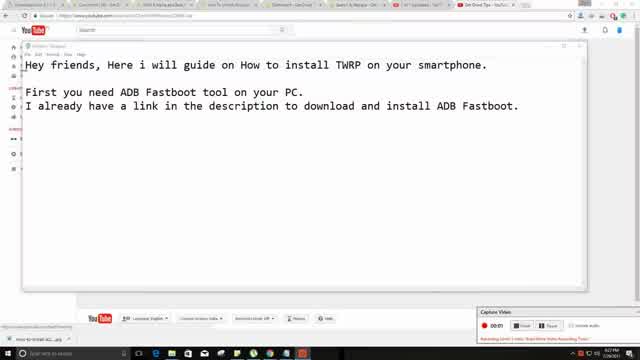
Flask is a web framework that provides libraries to build lightweight web applications in python. It is developed by **Armin Ronacher** who leads an international group of python enthusiasts (POCCO).

## What is Flask?

Flask is a web framework that provides libraries to build lightweight web applications in python. It is developed by **Armin Ronacher** who leads an international group of python enthusiasts (POCCO). It is based on WSGI toolkit and jinja2 template engine. Flask is considered as a micro framework.

## What is WSGI?

It is an acronym for web server gateway interface which is a standard for python web application development. It is considered as the specification for the universal interface between the web server and web application.



## What is Jinja2?

Jinja2 is a web template engine which combines a template with a certain data source to render the dynamic web pages.

## Flask Environment Setup

To install flask on the system, we need to have python 2.7 or higher installed on our system. However, we suggest using python 3 for the development in the flask.

### Install virtual environment (virtualenv)

virtualenv is considered as the virtual python environment builder which is used to create the multiple python virtual environment side by side. It can be installed by using the following command.

1. $ pip install virtualenv

Once it is installed, we can create the new virtual environment into a folder as given below.

1. $ mkdir new
2. $ cd new
3. $ virtualenv venv

To activate the corresponding environment, use the following command on the Linux operating system.

1. $ venv/bin/activate

On windows, use the following command.

1. $ venv\scripts\activate

We can now install the flask by using the following command.

1. $ pip install flask

However, we can install the flask using the above command without creating the virtual environment.

**Pycharm Tutorial**

PyCharm is the most popular IDE for Python, and includes great features such as excellent code completion and inspection with advanced debugger and support for web programming and various frameworks. PyCharm is created by Czech company, Jet brains which focusses on creating integrated development environment for various web development languages like JavaScript and PHP.

# **Audience**

This tutorial has been prepared for Python developers who focus on using IDE with complete package of running, debugging and creating projects in various python frameworks. Also, interested learners with a basic knowledge of any IDE can take up this tutorial.

# **Prerequisites**

Before proceeding with this tutorial, you need a basic knowledge of any integrated development environment of Python like Sublime Text or most popular IDE like NetBeans. If you are a beginner, we suggest you to go through tutorials related to these topics first before proceeding further on this tutorial.

PyCharm is the most popular IDE used for Python scripting language. This chapter will give you an introduction to PyCharm and explains its features.

PyCharm offers some of the best features to its users and developers in the following aspects −

* Code completion and inspection
* Advanced debugging
* Support for web programming and frameworks such as Django and Flask

## Features of PyCharm

Besides, a developer will find PyCharm comfortable to work with because of the features mentioned below −

### Code Completion

PyCharm enables smoother code completion whether it is for built in or for an external package.

### SQLAlchemy as Debugger

You can set a breakpoint, pause in the debugger and can see the SQL representation of the user expression for SQL Language code.

### Git Visualization in Editor

When coding in Python, queries are normal for a developer. You can check the last commit easily in PyCharm as it has the blue sections that can define the difference between the last commit and the current one.

### Code Coverage in Editor

You can run **.py** files outside PyCharm Editor as well marking it as code coverage details elsewhere in the project tree, in the summary section etc.

### Package Management

All the installed packages are displayed with proper visual representation. This includes list of installed packages and the ability to search and add new packages.

### Local History

Local History is always keeping track of the changes in a way that complements like Git.

Local history in PyCharm gives complete details of what is needed to rollback and what is to be added.

### Refactoring

Refactoring is the process of renaming one or more files at a time and PyCharm includes various shortcuts for a smooth refactoring process.

### User Interface of PyCharm Editor

The user interface of PyCharm editor is shown in the screenshot given below. Observe that the editor includes various features to create a new project or import from an existing project.

You can download the PyCharm Editor and read its official documentation at this link − <https://www.jetbrains.com/pycharm/>

# **SYSTEM DESIGN**

**UML DIAGRAMS**

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

**Global Use Case Diagrams:**

Identification of actors:

**Actor:** Actor represents the role a user plays with respect to the system. An actor interacts with, but has no control over the use cases.

Graphical representation:



<<Actor name>>

An actor is someone or something that:

Interacts with or uses the system.

* Provides input to and receives information from the system.
* Is external to the system and has no control over the use cases.

Actors are discovered by examining:

* Who directly uses the system?
* Who is responsible for maintaining the system?
* External hardware used by the system.
* Other systems that need to interact with the system.

Questions to identify actors:

* + Who is using the system? Or, who is affected by the system? Or, which groups need help from the system to perform a task?
  + Who affects the system? Or, which user groups are needed by the system to perform its functions? These functions can be both main functions and secondary functions such as administration.
  + Which external hardware or systems (if any) use the system to perform tasks?
  + What problems does this application solve (that is, for whom)?
  + And, finally, how do users use the system (use case)? What are they doing with the system?

The actors identified in this system are:

1. **System Administrator**
2. **Customer**
3. **Customer Care**

Identification of usecases:

**Usecase:** A use case can be described as a specific way of using the system from a user’s (actor’s) perspective.

**Graphical representation:**



A more detailed description might characterize a use case as:

* Pattern of behavior the system exhibits
* A sequence of related transactions performed by an actor and the system
* Delivering something of value to the actor

Use cases provide a means to:

* capture system requirements
* communicate with the end users and domain experts
* test the system

Use cases are best discovered by examining the actors and defining what the actor will be able to do with the system.

Guide lines for identifying use cases:

* For each actor, find the tasks and functions that the actor should be able to perform or that the system needs the actor to perform. The use case should represent a course of events that leads to clear goal
* Name the use cases.
* Describe the use cases briefly by applying terms with which the user is familiar.

This makes the description less ambiguous

Questions to identify use cases:

* What are the tasks of each actor?
* Will any actor create, store, change, remove or read information in the system?
* What use case will store, change, remove or read this information?
* Will any actor need to inform the system about sudden external changes?
* Does any actor need to inform about certain occurrences in the system?
* What usecases will support and maintains the system?

**Flow of Events**

A flow of events is a sequence of transactions (or events) performed by the system. They typically contain very detailed information, written in terms of what the system should do, not how the system accomplishes the task. Flow of events are created as separate files or documents in your favorite text editor and then attached or linked to a use case using the Files tab of a model element.

A flow of events should include:

* When and how the use case starts and ends
* Use case/actor interactions
* Data needed by the use case
* Normal sequence of events for the use case
* Alternate or exceptional flows

Construction of Usecase diagrams:

Use-case diagrams graphically depict system behavior (use cases). These diagrams present a high level view of how the system is used as viewed from an outsider’s (actor’s) perspective. A use-case diagram may depict all or some of the use cases of a system.

A use-case diagram can contain:

* actors ("things" outside the system)
* use cases (system boundaries identifying what the system should do)
* Interactions or relationships between actors and use cases in the system including the associations, dependencies, and generalizations.

Relationships in use cases:

**1. Communication:**

The communication relationship of an actor in a usecase is shown by connecting the actor symbol to the usecase symbol with a solid path. The actor is said to communicate with the usecase.

**2. Uses:**

A Uses relationship between the usecases is shown by generalization arrow from the usecase.

**3. Extends:**

The extend relationship is used when we have one usecase that is similar to another usecase but does a bit more. In essence it is like subclass.

**SEQUENCE DIAGRAMS**

A sequence diagram is a graphical view of a scenario that shows object interaction in a time-based sequence what happens first, what happens next. Sequence diagrams establish the roles of objects and help provide essential information to determine class responsibilities and interfaces.

There are two main differences between sequence and collaboration diagrams: sequence diagrams show time-based object interaction while collaboration diagrams show how objects associate with each other. A sequence diagram has two dimensions: typically, vertical placement represents time and horizontal placement represents different objects.

**Object:**

An object has state, behavior, and identity. The structure and behavior of similar objects are defined in their common class. Each object in a diagram indicates some instance of a class. An object that is not named is referred to as a class instance.

The object icon is similar to a class icon except that the name is underlined:

An object's concurrency is defined by the concurrency of its class.

**Message:**

A message is the communication carried between two objects that trigger an event. A message carries information from the source focus of control to the destination focus of control. The synchronization of a message can be modified through the message specification. Synchronization means a message where the sending object pauses to wait for results.

**Link:**

A link should exist between two objects, including class utilities, only if there is a relationship between their corresponding classes. The existence of a relationship between two classes symbolizes a path of communication between instances of the classes: one object may send messages to another. The link is depicted as a straight line between objects or objects and class instances in a collaboration diagram. If an object links to itself, use the loop version of the icon.

**CLASS DIAGRAM:**

Identification of analysis classes:

A class is a set of objects that share a common structure and common behavior (the same attributes, operations, relationships and semantics). A class is an abstraction of real-world items.

There are 4 approaches for identifying classes:

1. Noun phrase approach:
2. Common class pattern approach.
3. Use case Driven Sequence or Collaboration approach.
4. Classes , Responsibilities and collaborators Approach
5. **Noun Phrase Approach:**

The guidelines for identifying the classes:

* + Look for nouns and noun phrases in the usecases.
  + Some classes are implicit or taken from general knowledge.
  + All classes must make sense in the application domain; Avoid computer

implementation classes – defer them to the design stage.

* + Carefully choose and define the class names After identifying the classes we have to eliminate the following types of classes:
* Adjective classes.

1. **Common class pattern approach:**

The following are the patterns for finding the candidate classes:

* + Concept class.
  + Events class.
  + Organization class
  + Peoples class
  + Places class
  + Tangible things and devices class.

1. **Use case driven approach:**

We have to draw the sequence diagram or collaboration diagram. If there is need for some classes to represent some functionality then add new classes which perform those functionalities.

1. **CRC approach:**

The process consists of the following steps:

* + Identify classes’ responsibilities ( and identify the classes )
  + Assign the responsibilities
  + Identify the collaborators.

Identification of responsibilities of each class:

The questions that should be answered to identify the attributes and methods of a class respectively are:

1. What information about an object should we keep track of?
2. What services must a class provide?

Identification of relationships among the classes:

Three types of relationships among the objects are:

Association: How objects are associated?

Super-sub structure: How are objects organized into super classes and sub classes?

Aggregation: What is the composition of the complex classes?

Association:

The **questions** that will help us to identify the associations are:

1. Is the class capable of fulfilling the required task by itself?
2. If not, what does it need?
3. From what other classes can it acquire what it needs?

Guidelines for identifying the tentative associations:

* A dependency between two or more classes may be an association. Association often corresponds to a verb or prepositional phrase.
* A reference from one class to another is an association. Some associations are implicit or taken from general knowledge.

Some common association patterns are:

Location association like part of, next to, contained in…..

Communication association like talk to, order to ……

We have to eliminate the unnecessary association like implementation associations, ternary or n-ary associations and derived associations.

Super-sub class relationships:

Super-sub class hierarchy is a relationship between classes where one class is the parent class of another class (derived class).This is based on inheritance.

Guidelines for identifying the super-sub relationship, a generalization are

1***.* Top-down*:***

Look for noun phrases composed of various adjectives in a class name. Avoid excessive refinement. Specialize only when the sub classes have significant behavior.

2.**Bottom-up*:***

Look for classes with similar attributes or methods. Group them by moving the common attributes and methods to an abstract class. You may have to alter the definitions a bit.

3.**Reusability*:***

Move the attributes and methods as high as possible in the hierarchy.

4. **Multiple inheritances*:***

Avoid excessive use of multiple inheritances. One way of getting benefits of multiple inheritances is to inherit from the most appropriate class and add an object of another class as an attribute.

**Aggregation or a-part-of relationship:**

It represents the situation where a class consists of several component classes. A class that is composed of other classes doesn’t behave like its parts. It behaves very difficultly. The major properties of this relationship are transitivity and anti symmetry.

The **questions** whose answers will determine the distinction between the part and whole relationships are:

* Does the part class belong to the problem domain?
* Is the part class within the system’s responsibilities?
* Does the part class capture more than a single value?( If not then simply include it as an attribute of the whole class)
* Does it provide a useful abstraction in dealing with the problem domain?

There are three types of aggregation relationships. They are:

**Assembly:**

It is constructed from its parts and an assembly-partsituation physically exists.

**Container:**

A physical whole encompasses but is not constructed from physical parts.

**Collection member:**

A conceptual whole encompasses parts that may be physical or conceptual. The container and collection are represented by hollow diamonds but composition is represented by solid diamond.

## **5.1 UML Diagrams**

****

Figure.5.1 Use case Diagram

****

Figure.5.2 Sequence Diagram for Admin

****

Figure.5.3 Sequence Diagram for User

****

Figure.5.4 State Chart Diagram

****

Figure.5.5 Activity Diagram

****

Figure.5.6 Component Diagram

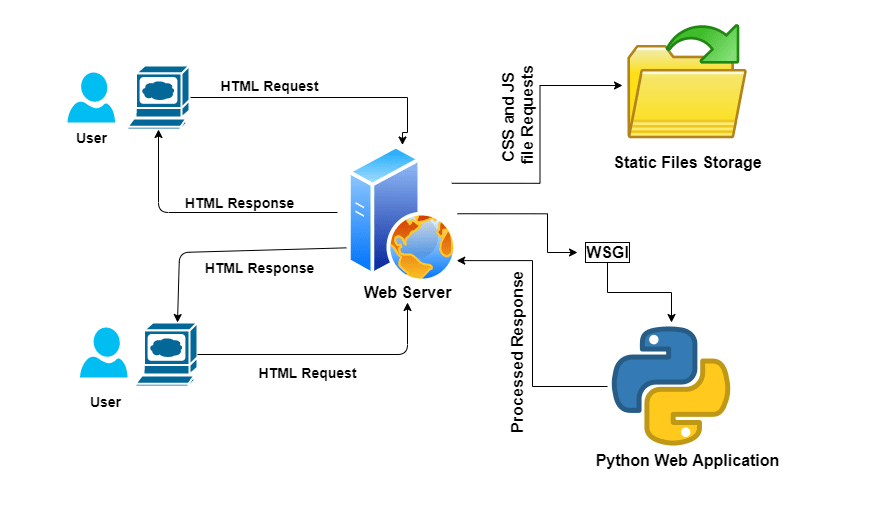


Figure.5.7 Deployment Diagram

**About MySQL:**

**MySQL** is a relational database management system (RDBMS)[]](http://en.wikipedia.org/wiki/MySQL#cite_note-1) that runs as a server providing multi-user access to a number of databases.  The SQL phrase stands for Structured Query Language.Free-software-open source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla, WordPress, phpBB, Drupal and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google  , Facebook, and Twitter.

MySQL is the world's most popular open source database software, with over 100 million copies of its software downloaded or distributed throughout it's history. With its superior speed, reliability, and ease of use, MySQL has become the preferred choice for Web, Web 2.0, SaaS, ISV, Telecom companies and forward-thinking corporate IT Managers because it eliminates the major problems associated with downtime, maintenance and administration for modern, online applications.

Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, critical business systems, and packaged software including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, Wikipedia, and Booking.com.

The flagship MySQL offering is MySQL Enterprise, a comprehensive set of production-tested software, proactive monitoring tools, and premium support services available in an affordable annual subscription.

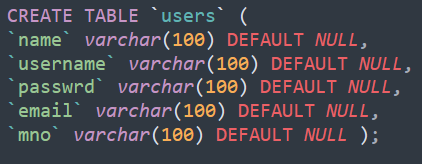
MySQL is a key part of LAMP (Linux, Apache, MySQL, PHP / Perl / Python), the fast-growing open source enterprise software stack. More and more companies are using LAMP as an alternative to expensive proprietary software stacks because of its lower cost and freedom from platform lock-in.

MySQL was originally founded and developed in Sweden by two Swedes and a Finn: David Axmark, Allan Larsson and Michael "Monty" Widenius, who had worked together since the 1980's. More historical information on MySQL is

## **Database Design**

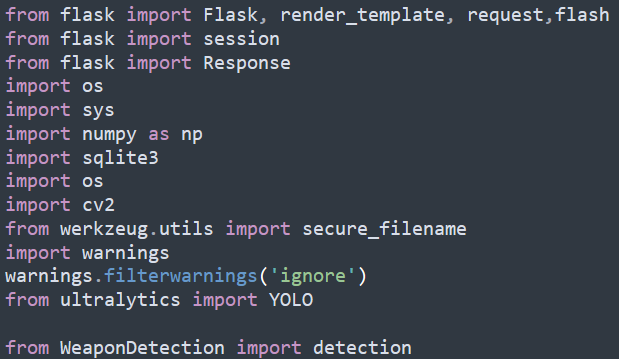
**Table name:** users

**Table Description:**



## **Sample Code**

**Imported Libraries:**



**dbconfig:**



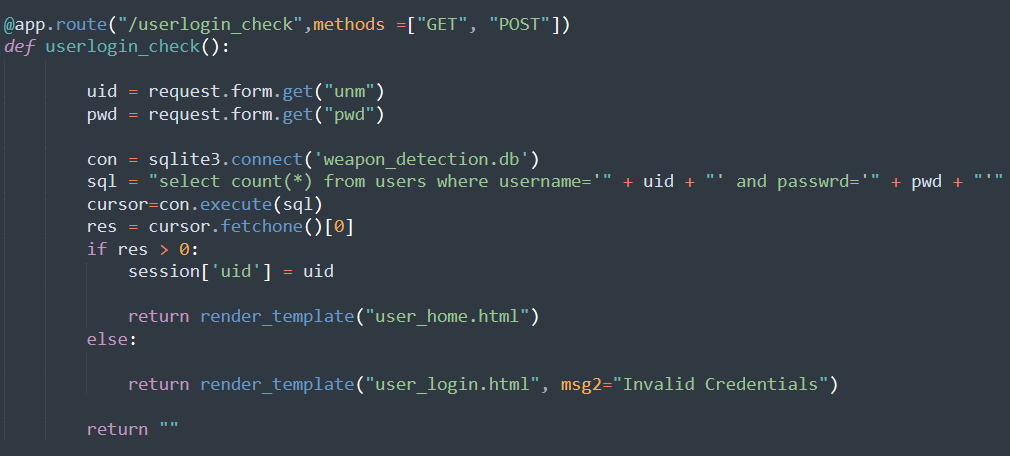
**adminLoginCheck:**



**User Registration:**

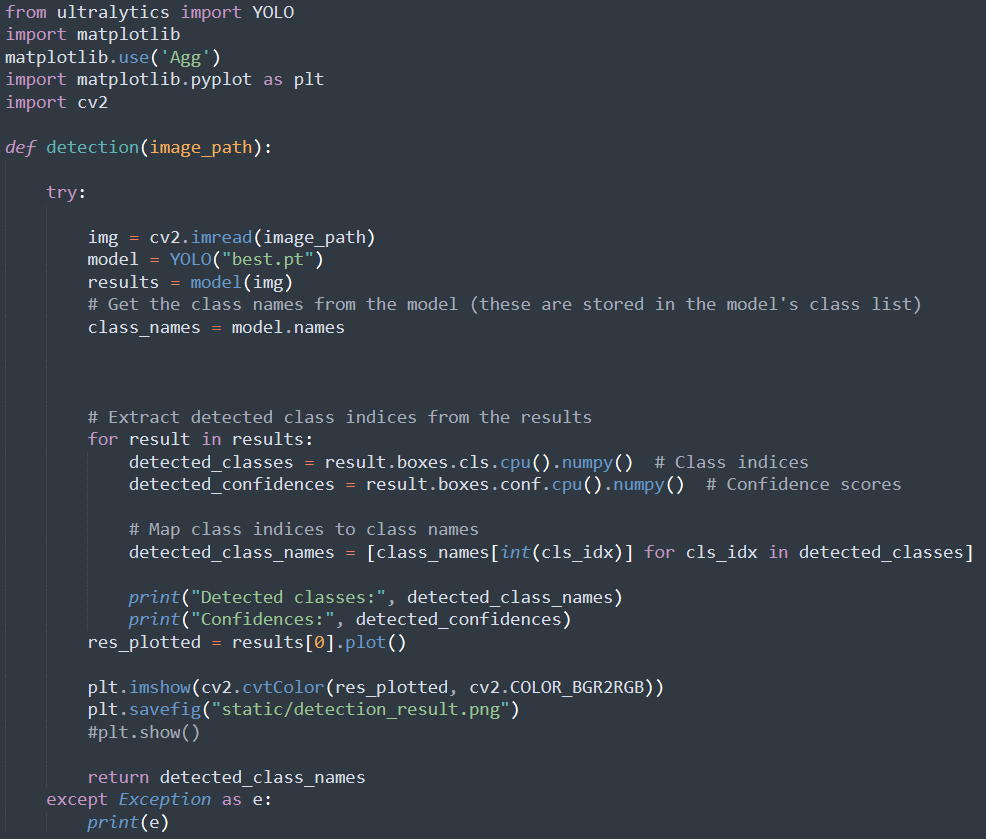


**User Login Check:**

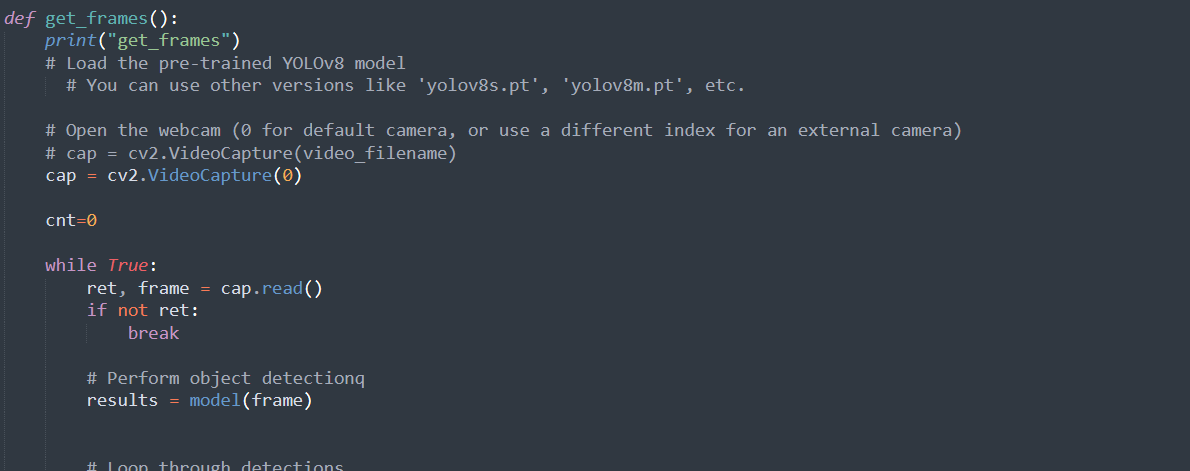
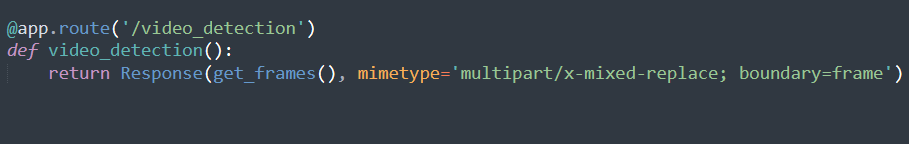


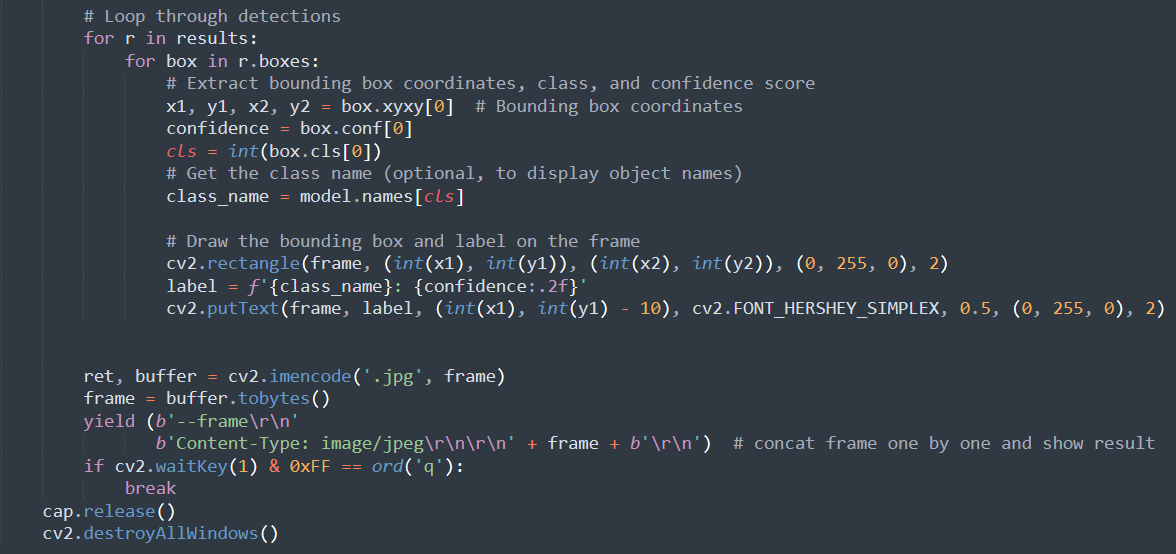
**Weapon Detection:**





**Live Detection:**





**SOFTWARE TESTING**

Software testing is one of the main stages of project development life cycle to provide our cessation utilizer with information about the quality of the application and ours, in our Project we have under gone some stages of testing like unit testing where it’s done in development stage of the project when we are in implementation of the application after the Project is yare we have done manual testing with different Case of all the different modules in the application we have even done browser compatibility testing in different web browsers in market, even we have done Client side validation testing on our application

**Unit testing**

The unit testing is done in the stage of implementation of the project only the error are solved in development stage some of the error we come across in development are given below

**TESTING**

Testing is the debugging program is one of the most critical aspects of the computer programming triggers, without programming that works, the system would never produce an output of which it was designed. Testing is best performed when user development is asked to assist in identifying all errors and bugs. The sample data are used for testing. It is not quantity but quality of the data used the matters of testing. Testing is aimed at ensuring that the system was accurately an efficiently before live operation commands.

**Testing objectives:**

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say, testing is a process of executing a program with intent of finding an error.

A successful test is one that uncovers an as yet undiscovered error.

A good test case is one that has probability of finding an error, if it exists.

The test is inadequate to detect possibly present errors.

The software more or less confirms to the quality and reliable standards.3999**Levels of Testing:**

In order to uncover present in different phases we have the concept of levels of testing.

**The basic levels of Testing:**

Client needs acceptance testing

Requirements system testing

Design integration testing

Code unit testing

Figure: Levels of Testing

**Code testing:**

This examines the logic of the program. For example, the logic for updating various sample data and with the sample files and directories were tested and verified.

**Specification Testing:**

Executing this specification starting what the program should do and how it should performed under various conditions. Test cases for various situation and combination of conditions in all the modules are tested.

**Unit testing:**

In the unit testing we test each module individually and integrate with the overall system. Unit testing focuses verification efforts on the smallest unit of software design in the module. This is also known as module testing. The module of the system is tested separately. This testing is carried out during programming stage itself. In the testing step each module is found to work satisfactorily as regard to expected output from the module. There are some validation checks for fields also. For example the validation check is done for varying the user input given by the user which validity of the data entered. It is very easy to find error debut the system.

Each Module can be tested using the following two Strategies:

1. Black Box Testing
2. White Box Testing

**BLACK BOX TESTING**

**What is Black Box Testing?**

Black box testing is a software testing techniques in which **functionality of the software under test (SUT) is tested without looking at the internal code structure**, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on the software requirements and specifications.

**In Black Box Testing we just focus on inputs and output of the software system** without bothering about internal knowledge of the software program.



The above Black Box can be any software system you want to test. For example : an operating system like Windows, a website like Google ,a database like Oracle or even your own custom application. Under Black Box Testing, you can test these applications by just focusing on the inputs and outputs without knowing their internal code implementation.

**Black box testing - Steps**

Here are the generic steps followed to carry out any type of Black Box Testing.

* Initially requirements and specifications of the system are examined.
* Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly. Also some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.
* Tester determines expected outputs for all those inputs.
* Software tester constructs test cases with the selected inputs.
* The test cases are executed.
* Software tester compares the actual outputs with the expected outputs.
* Defects if any are fixed and re-tested.

**Types of Black Box Testing**

There are many types of Black Box Testing but following are the prominent ones -

* **Functional testing** – This black box testing type is related to functional requirements of a system; it is done by software testers.
* **Non-functional testing** – This type of black box testing is not related to testing of a specific functionality, but non-functional requirements  such as performance, scalability, usability.
* **Regression testing** – Regression testing is done  after code fixes , upgrades or any other system maintenance to check the new code has not affected the existing code.

**WHITE BOX TESTING**

White Box Testing is the testing of a software solution's internal coding and infrastructure.It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability.White box testing is also known as **clear, open, structural, and glass box testing**.

It is one of two parts of the **"box testing" approach** of software testing. Its counter-part, blackbox testing, involves testing from an external or end-user type perspective. On the other hand, Whitebox testing is based on the inner workings of an application and revolves around internal testing. The term "whitebox" was used because of the see-through box concept. The clear box or whitebox name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "black box testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested

## What do you verify in White Box Testing ?

White box testing involves the testing of the software code for the following:

* Internal security holes
* Broken or poorly structured paths in the coding processes
* The flow of specific inputs through the code
* Expected output
* The functionality of conditional loops
* Testing of each statement, object and function on an individual basis

The testing can be done at system, integration and unit levels of software development. One of the basic goals of whitebox testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected or desired outputs so that when a specific input does not result in the expected output, you have encountered a bug.

**How do you perform White Box Testing?**

To give you a simplified explanation of white box testing, we have divided it into **two basic steps**. This is what testers do when testing an application using the white box testing technique:

**STEP 1) UNDERSTAND THE SOURCE CODE**

The first thing a tester will often do is learn and understand the source code of the application. Since white box testing involves the testing of the inner workings of an application, the tester must be very knowledgeable in the programming languages used in the applications they are testing. Also, the testing person must be highly aware of secure coding practices. Security is often one of the primary objectives of testing software. The tester should be able to find security issues and prevent attacks from hackers and naive users who might inject malicious code into the application either knowingly or unknowingly.

**Step 2) CREATE TEST CASES AND EXECUTE**

The second basic step to white box testing involves testing the application’s source code for proper flow and structure. One way is by writing more code to test the application’s source code. The tester will develop little tests for each process or series of processes in the application. This  method requires that the tester must have intimate knowledge of the code and is often done by the developer. Other methods include manual testing, trial and error testing and the use of testing tools as we will explain further on in this article.

**System testing:**

Once the individual module testing is completed, modules are assembled and integrated to perform as a system. The top down testing, which began from upper level to lower level module, was carried out to check whether the entire system is performing satisfactorily.

There are three main kinds of System testing:

1. Alpha Testing
2. Beta Testing
3. Acceptance Testing

**Alpha Testing:**

This refers to the system testing that is carried out by the test team with the Organization.

**Beta Testing**:

This refers to the system testing that is performed by a selected group of friendly customers

**Acceptance Testing:**

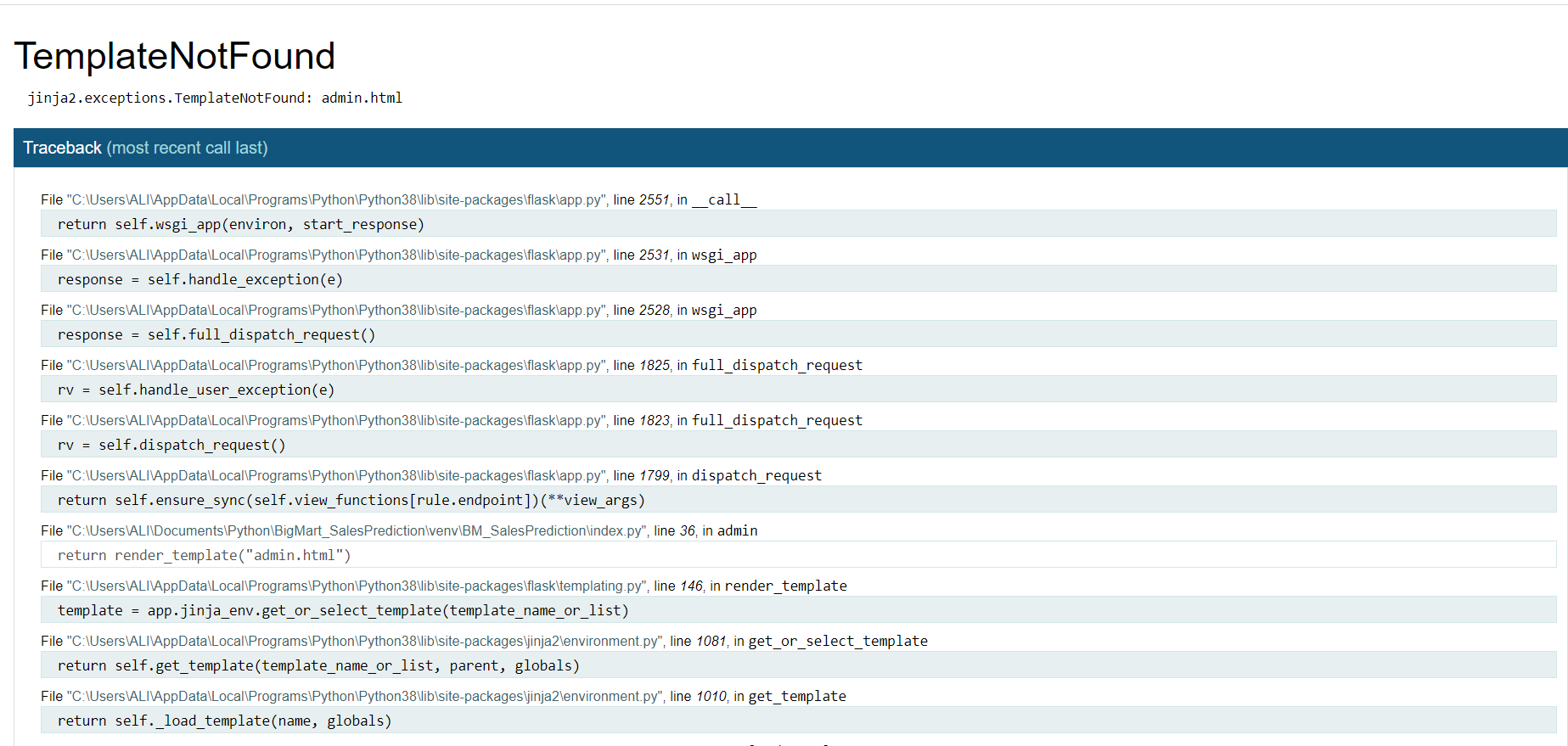
This refers to the system testing that is performed by the customer to determine whether or not to accept the delivery of the system.

**Integration Testing:**

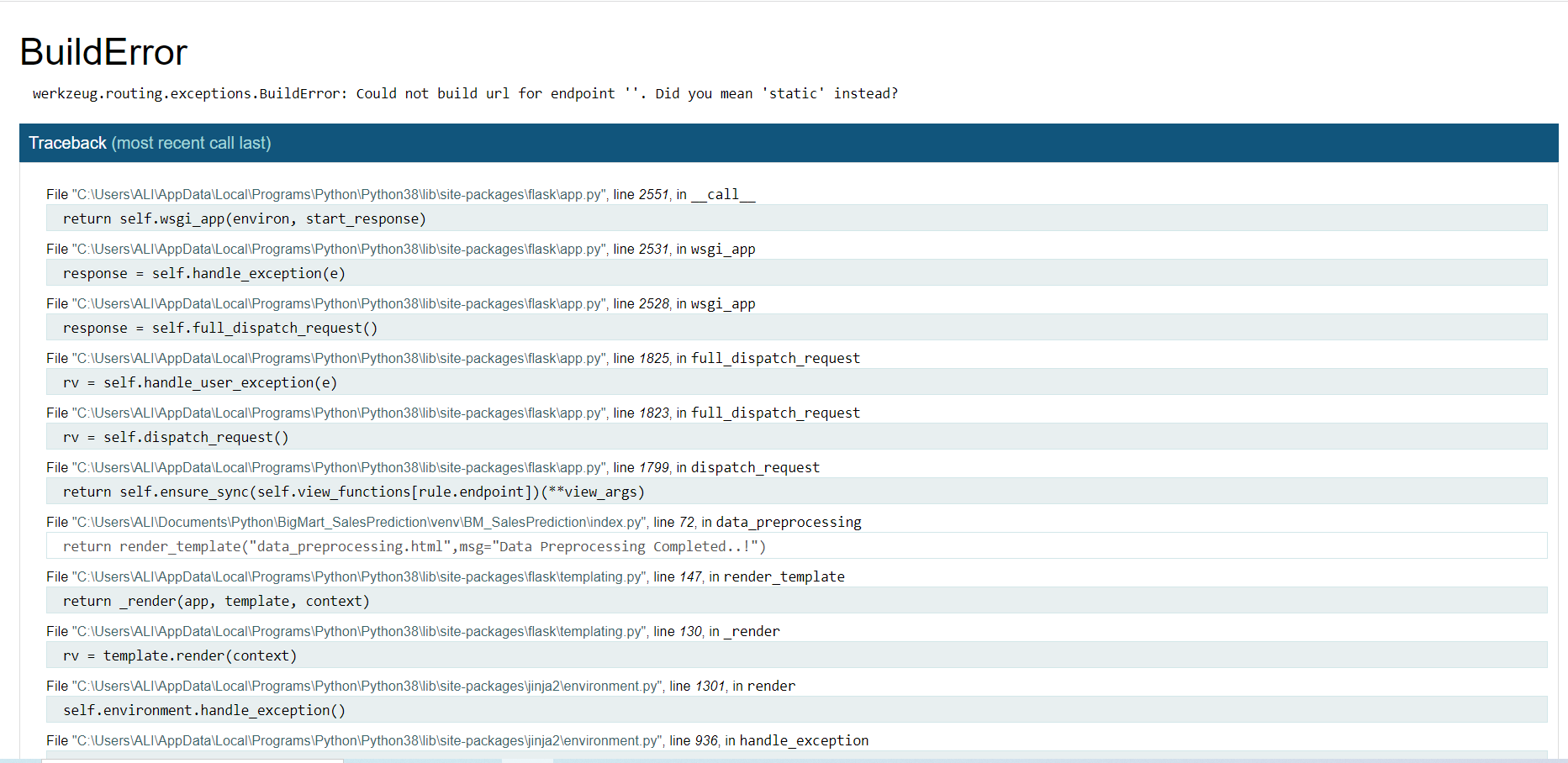
Data can be lost across an interface, one module can have an adverse effort on the other sub functions, when combined, may not produce the desired major functions. Integrated testing is the systematic testing for constructing the uncover errors within the interface. The testing was done with sample data. The developed system has run successfully for this sample data. The need for integrated test is to find the overall system performance.

**Output testing:**After performance of the validation testing, the next step is output testing. The output displayed or generated by the system under consideration is tested by asking the user about the format required by system.

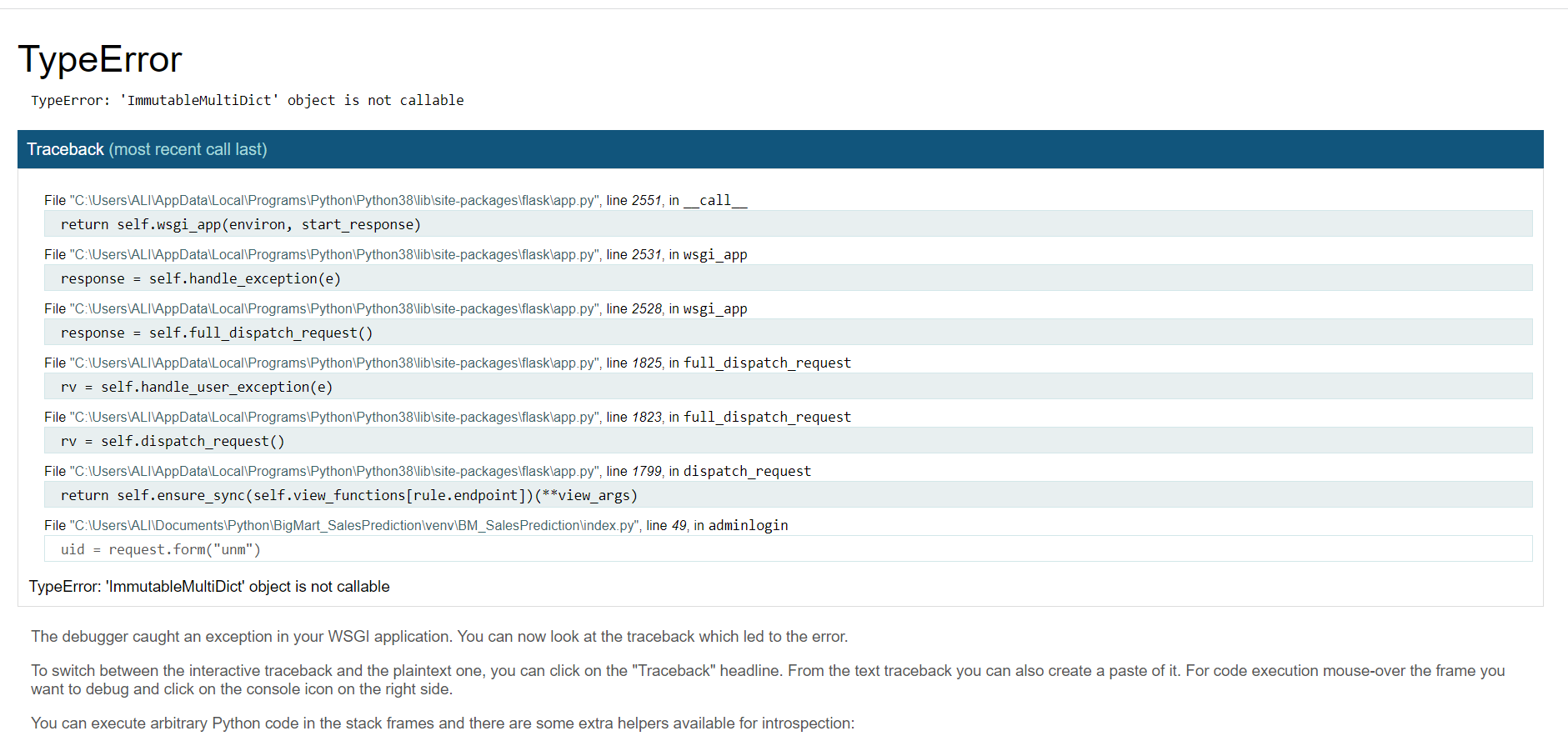
**Unit testing**



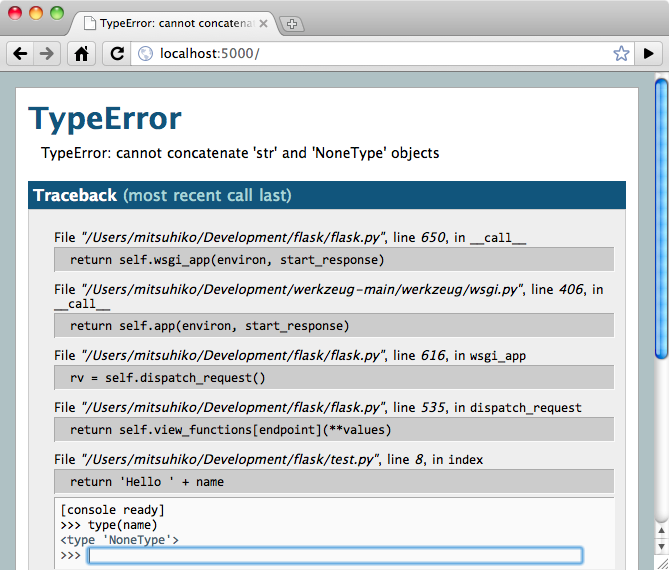
Template Not four Error



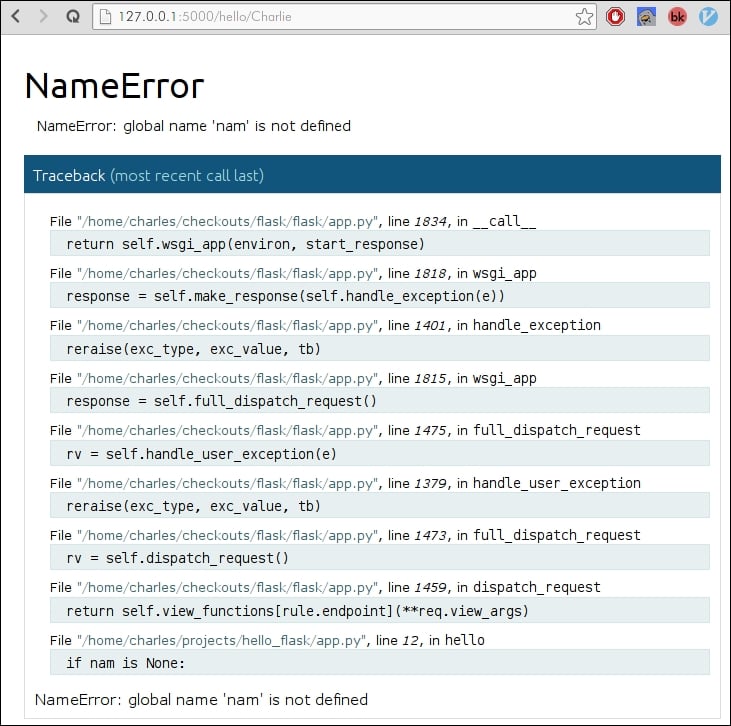
Build model missing error



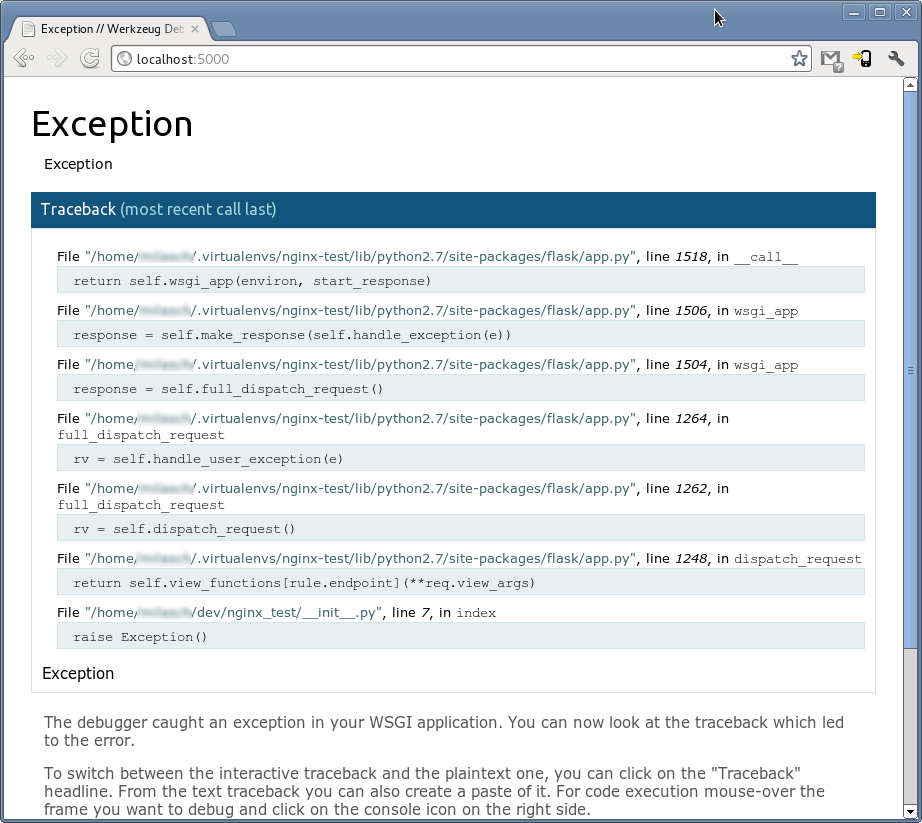
**TypeError Testing:**



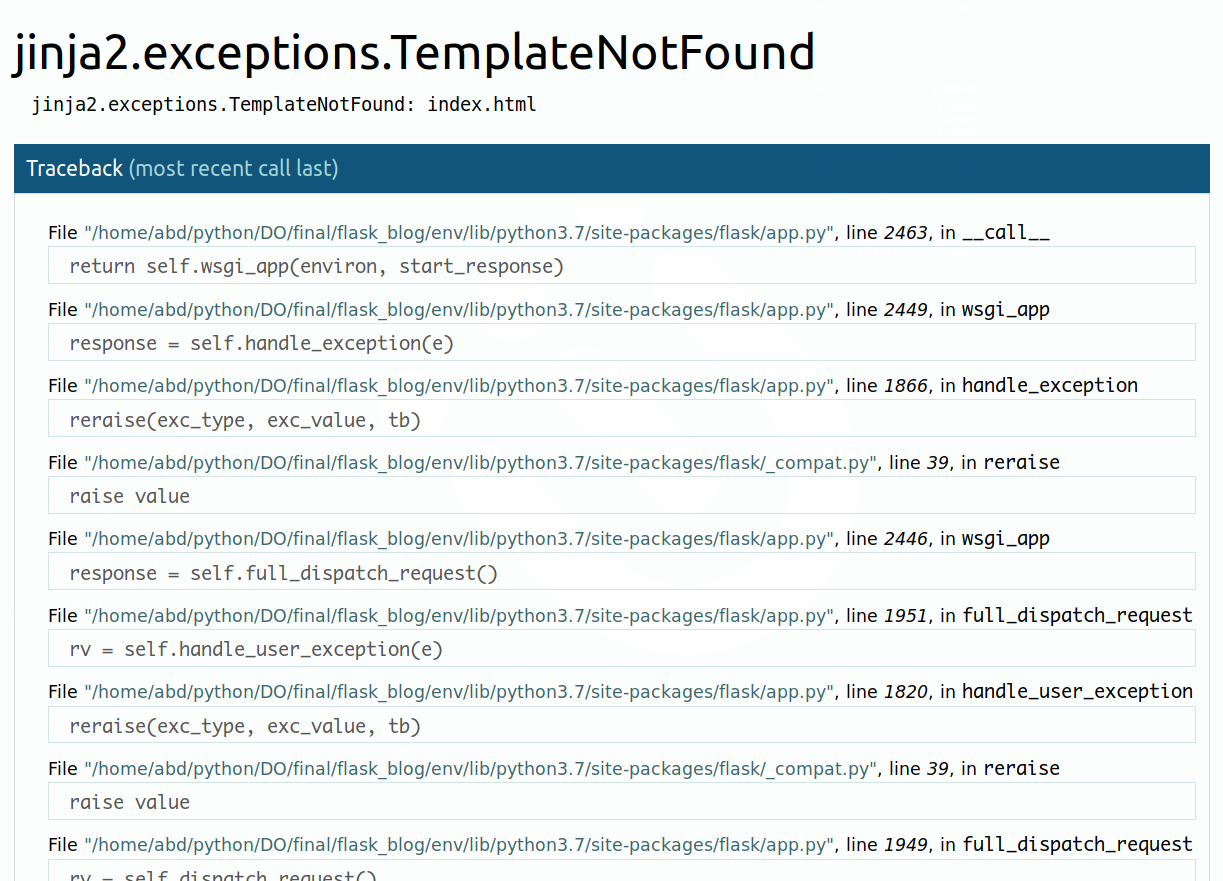
**Name Error Testing:**



**Exception:**

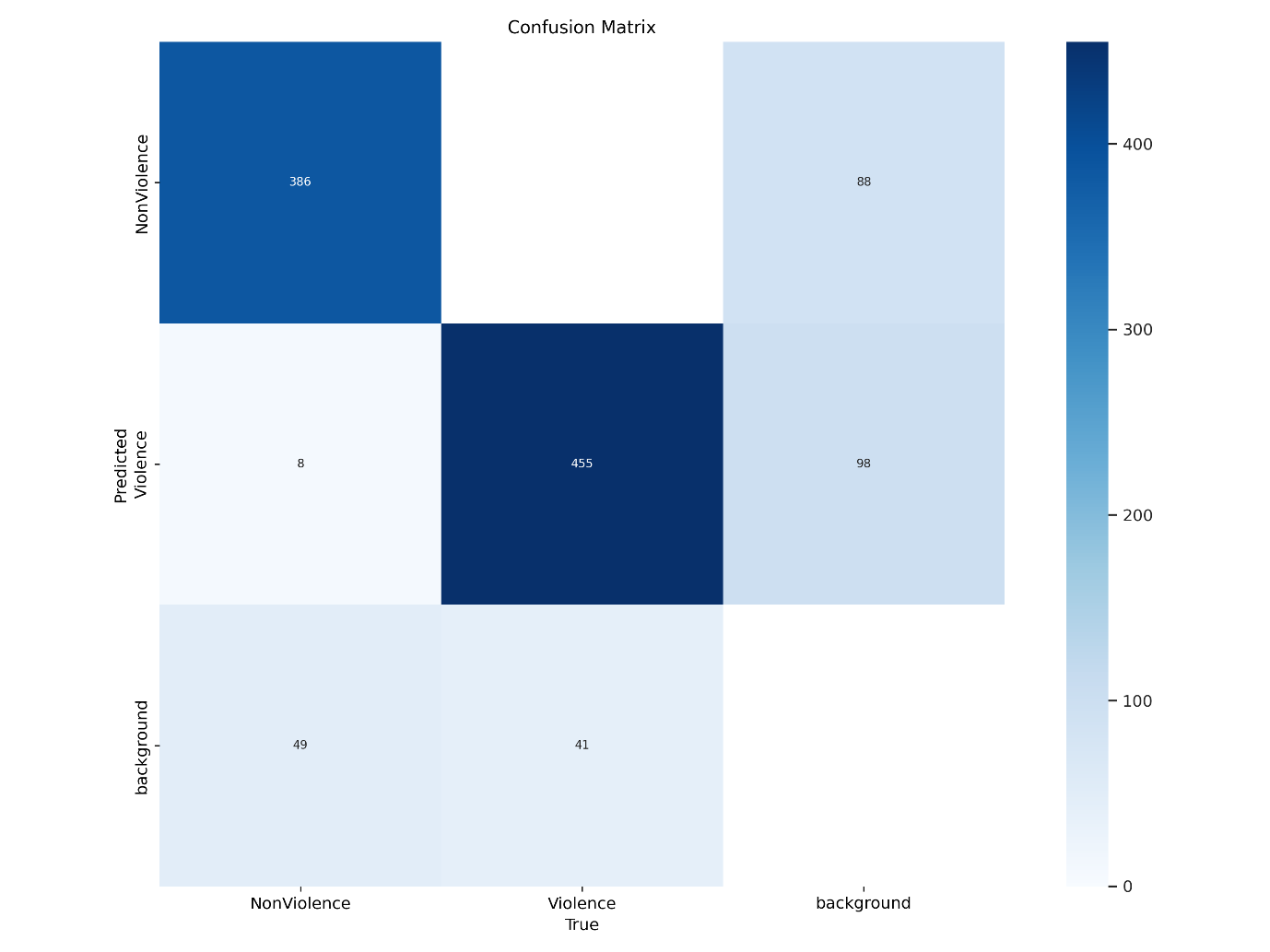


**TemplateNotFound:**

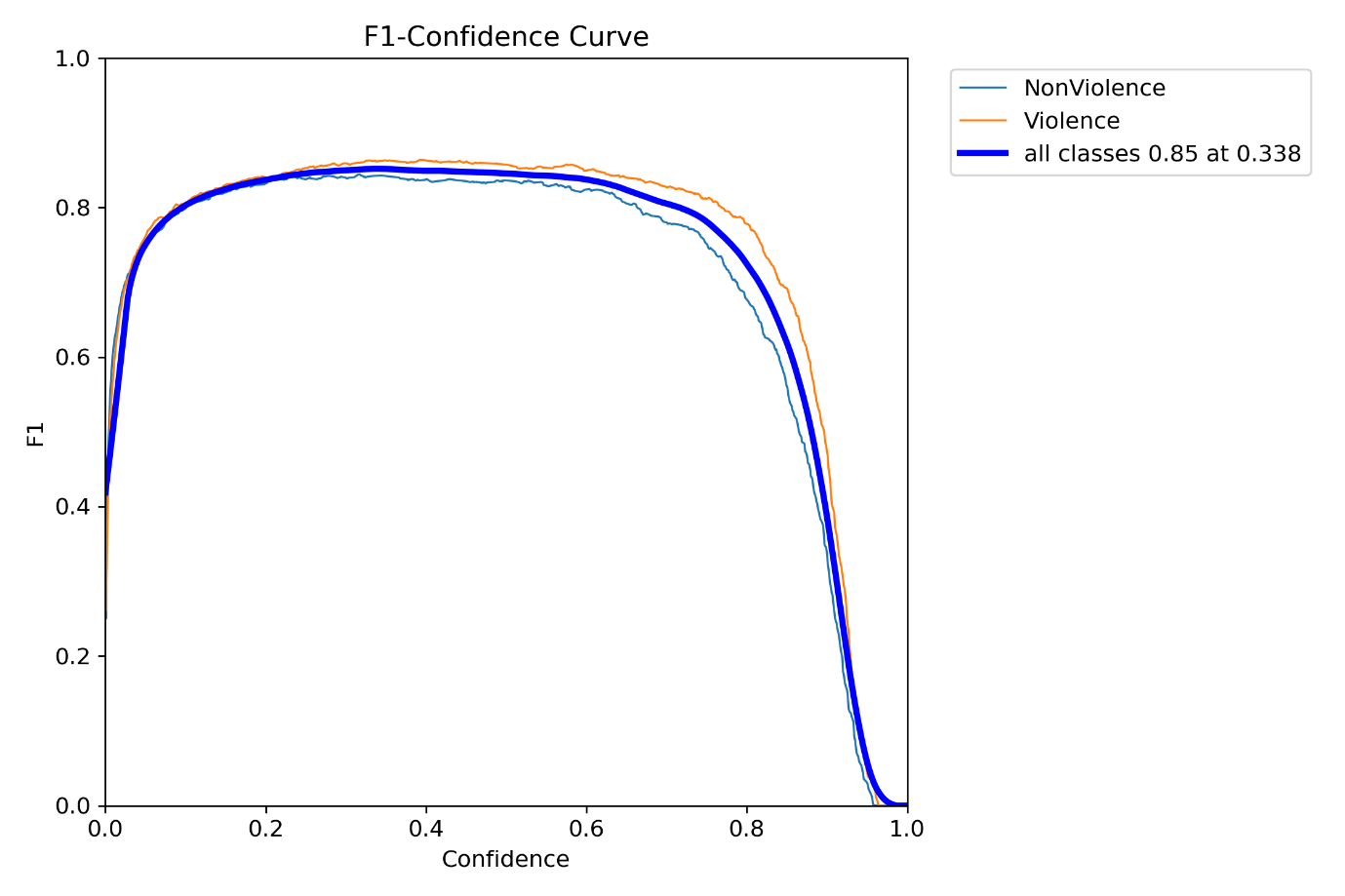


**SCREEN SHORTS**

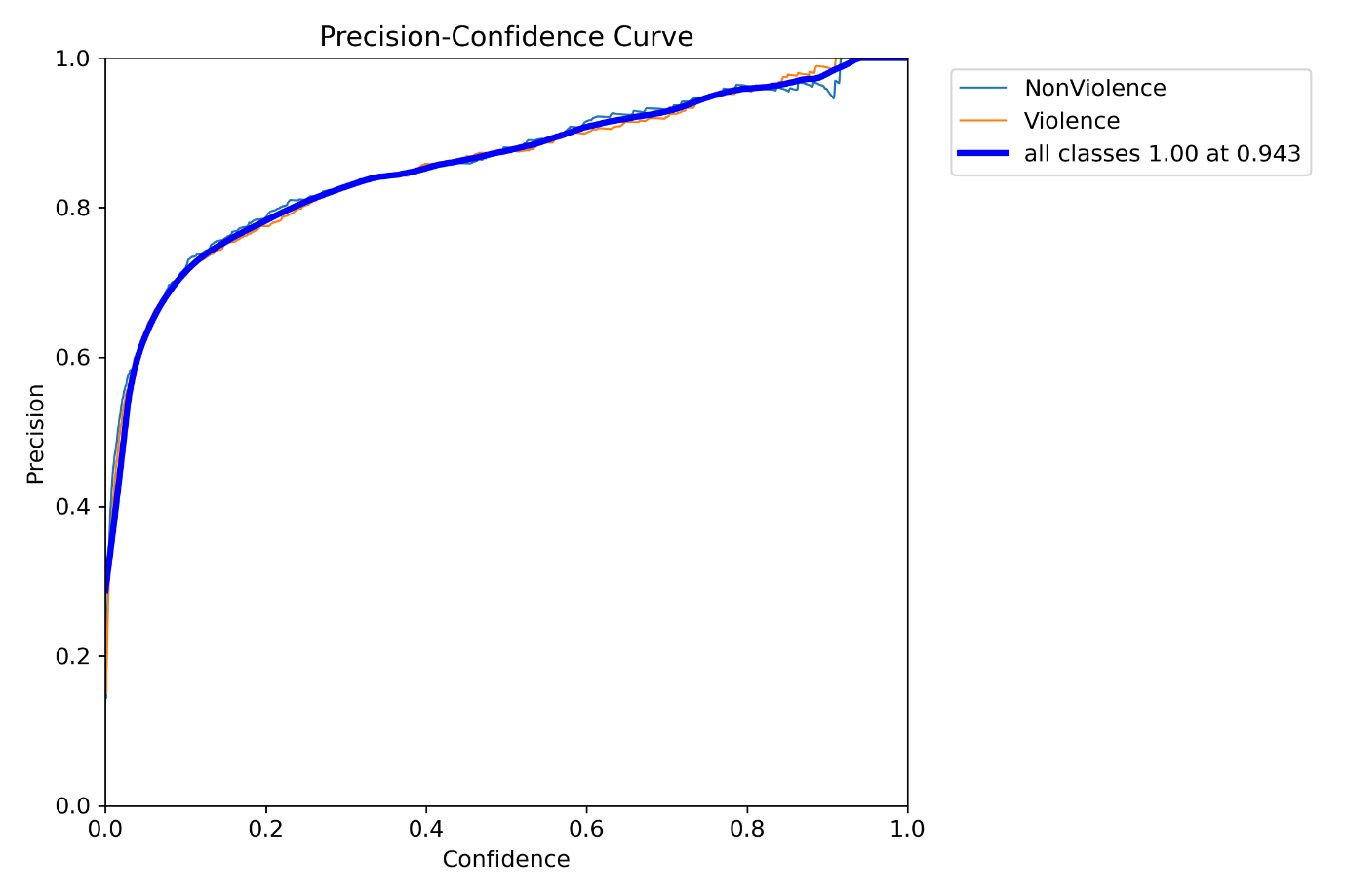
**confusion\_matrix**

****

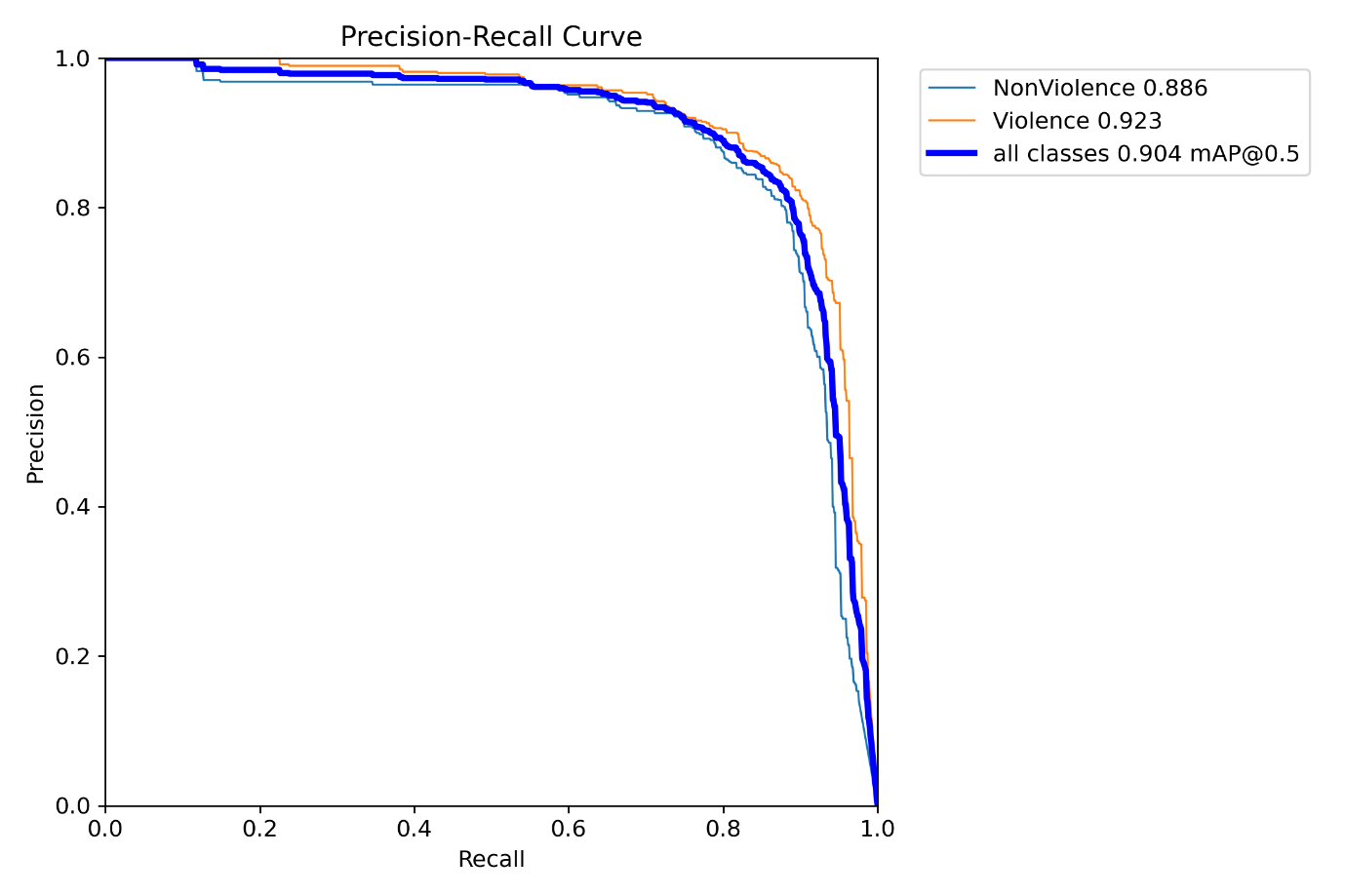
**F1\_curve**

****

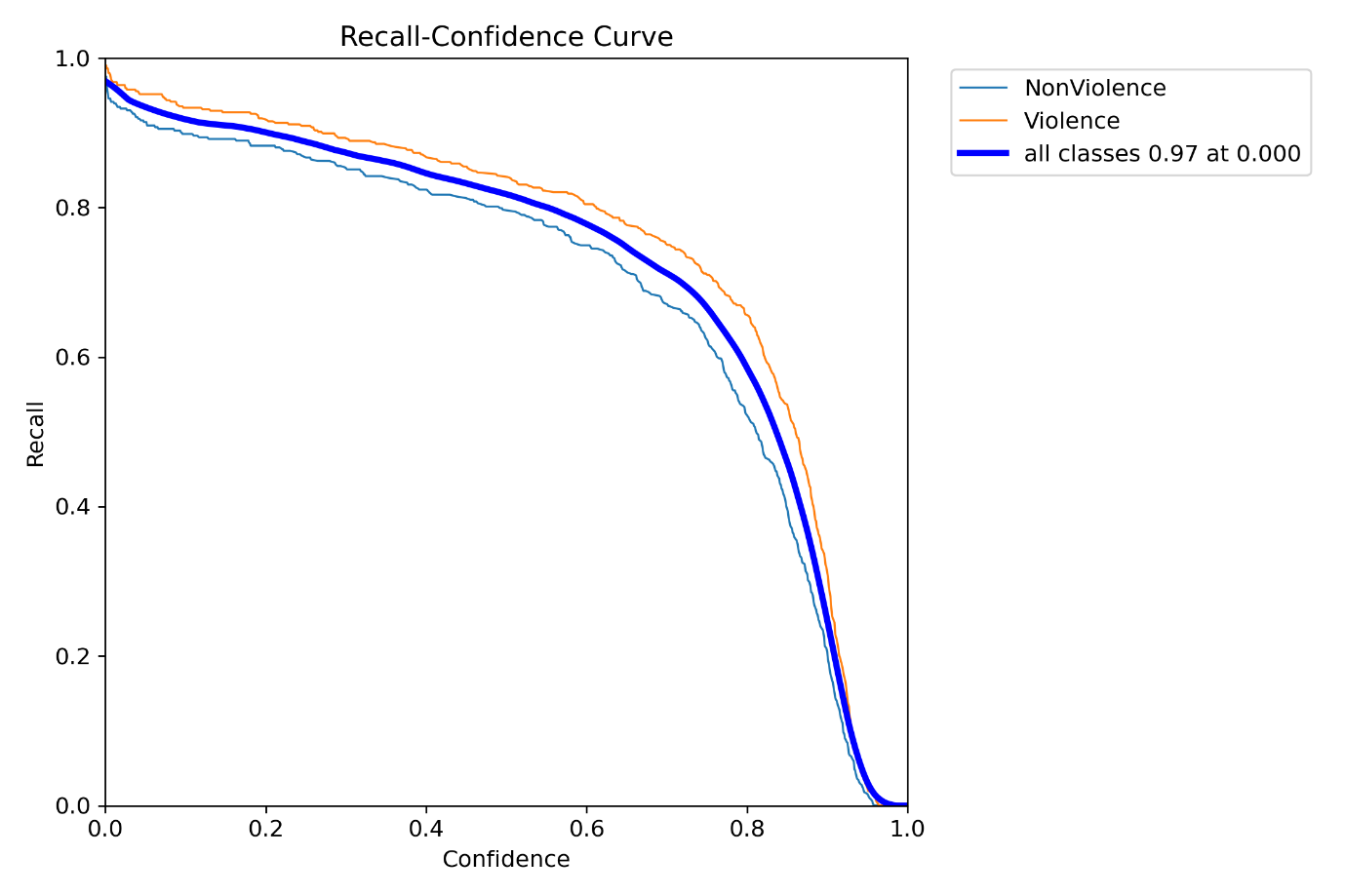
**P\_curve**

****

**PR\_curve**

****

**R\_curve**

****

# **CONCLUSION**

The rapid advancement of computer vision technologies has significantly improved the efficiency and reliability of surveillance systems in ensuring public safety. This project demonstrates the potential of using YOLOv8, a state-of-the-art convolutional neural network (CNN), for real-time weapon detection. By leveraging the Roboflow weapon detection dataset, the system achieves high accuracy in identifying and classifying weapons in live video feeds or images. This capability is crucial for enhancing security measures, particularly in high-risk or crowded environments where the threat of violence may escalate. The integration of YOLOv8 into surveillance systems offers a scalable and effective solution for anomaly detection and situational analysis. This research not only contributes to addressing critical challenges in public safety but also serves as a foundation for future developments in automated monitoring and response systems.

# **10. REFERENCES**

[1] H. Jain, A. Vikram, Mohana, A. Kashyap, and A. Jain, “Weapon Detection using Artificial Intelligence and Deep Learning for Security Applications,” 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020, pp. 10.1109/ICESC48915.2020.9155832. 193-198, doi: 10.1109/ICESC48915.2020.9155832.

[2] P.Shanmugapriya.Gurram, Reddy,J Kumar “Weapon Detection using Artificial Intelligence and Deep Learning for Security Applications,” Mar 2022 International Research Journal of Engineering and Technology (IRJET) Volume: 09 Issue: 03.

[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 Rohitet.al., “A Review of Artificial Intelligence Methods for Data Science and Data Analytics: Applications and Research Challenges,”2018 2nd International Conference 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 Technology,2018.

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

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

[8] Prof.Vishal R. Shinde," An application of deep learning algorithm for automatic detection of unexpected accidents under bad CCTV” in IJREAM, ISSN: 2454-9150, Volume 08, Issue 01, APR 2022 Special Issue.