# Efficient Railway Foreign Object Detection Using Enhanced YOLO with Efficient Net Backbone and Attention Mechanisms

**Author 1: D. srihitha**
Email: srihithareddydandu@gmail.com

**Author 2: B. Naveen**
Email: naveengoud898@gmail.com

**Author 3: CH.Abhinav**
Email: 22wj1a0558@gniindia.org

**Guide: Ms.R.Santhoshi**
**Organization:** Guru Nanak Institutions, India

**keywords:** Railway Safety, foreign object detection, deep learning , real time monitoring , railway surveillance.

**ABSTRACT:**
Ensuring the timely and accurate detection of foreign objects on railway tracks is vital for maintaining the safety and reliability of rail transport systems. This study presents an enhanced foreign object intrusion detection framework that addresses the limitations of existing methods—namely low efficiency and suboptimal accuracy—by integrating a two-stage architecture based on YOLOv8 and Overhaul Knowledge Distillation (OKD). In the first stage, a lightweight image classification model rapidly filters railway images to identify those potentially containing foreign objects, reducing the computational burden on detection models. Images flagged as suspicious are then passed to the second stage, where the YOLOv8 object detector precisely localizes and identifies the foreign objects. The use of YOLOv8 offers significant gains in both detection accuracy and inference speed over its predecessors. To further boost the performance of the classification stage, the Overhaul Knowledge Distillation technique is employed, allowing the lightweight classifier to learn from a more complex teacher network and achieve competitive accuracy with improved efficiency. Experimental evaluations confirm that the proposed approach outperforms existing solutions in both speed and robustness, establishing a new state-of

## CHAPTER-1

## INTRODUCTION

1

1.Introduction :

Detecting foreign objects on railway tracks is critical to ensuring the safety and uninterrupted operation of train systems. Obstructions such as debris, rocks, or fallen branches can lead to serious accidents, derailments, and infrastructure damage—resulting in both safety hazards and economic losses. Traditional detection methods, including manual inspections and classical image processing techniques, often fail to meet the demands of real-time operation, especially under varying weather or lighting conditions. Older deep learning models like YOLOv3 and YOLOv5 have also shown limitations in handling small or irregular objects and require high computational resources for deployment. To overcome these challenges, this project proposes a two-stage foreign object detection framework leveraging YOLOv8 and Overhaul Knowledge Distillation (OKD). In the first stage, a lightweight classification model rapidly filters out negative samples to reduce computational load. The second stage employs YOLOv8, a powerful and efficient object detection algorithm, to accurately detect and localize foreign objects in the flagged images. YOLOv8's anchor-free architecture and improved speed-accuracy balance make it ideal for real-time applications. Additionally, the integration of OKD enables the lightweight classifier to learn effectively from a larger teacher model, maintaining high accuracy while ensuring fast inference—making the overall system both scalable and efficient for practical railway monitoring scenarios.

## 1.2 SCOPE OF THE PROJECT

The scope of this project is to design and implement an efficient, accurate, and real-time system for detecting foreign objects on railway tracks using advanced deep learning techniques. The system focuses on identifying a wide range of obstructions such as debris, rocks, equipment, and fallen objects that pose safety risks to railway operations. It aims to improve upon traditional detection methods by reducing manual effort, increasing detection speed, and maintaining high accuracy under diverse environmental conditions. This project includes the development of a two-stage detection framework combining a lightweight image classification model with the YOLOv8 object detection algorithm. The solution is optimized for real-time performance and is suitable for deployment on surveillance systems, edge devices, or embedded hardware installed along railway tracks. The system also integrates Overhaul Knowledge Distillation to train the lightweight classifier efficiently without compromising accuracy. Overall, the project contributes to building a reliable, scalable, and automated solution for railway safety and monitoring applications.

## 1.3 OBJECTIVE

The primary objective of this project is to develop a robust and intelligent system for the detection of foreign objects on railway tracks using deep learning. The system is designed to enhance railway safety by identifying potential obstructions—such as debris, equipment, or natural obstacles—in real time, minimizing the risk of accidents and service interruptions. By leveraging the advanced capabilities of YOLOv8, the project aims to improve object detection accuracy, especially for small or irregularly shaped items that traditional methods often miss. Another key objective is to reduce computational load and improve processing efficiency through a two-stage detection approach. This includes the integration of a lightweight classification network to filter irrelevant images, combined with Overhaul Knowledge Distillation to maintain high accuracy without requiring heavy computation. The end goal is to build a fast, accurate, and deployable solution that works reliably under varying environmental conditions and can be implemented in real-world railway surveillance systems.

## 1.4 EXISTING SYSTEM:

Railway foreign object detection has traditionally depended on manual inspections and fixed sensor-based systems, such as infrared or ultrasonic sensors. While these methods are widely used, they come with significant drawbacks: they are expensive to deploy and maintain, require considerable human labor, and their performance is heavily affected by adverse weather, lighting, and visibility conditions. These limitations pose a serious threat to railway safety, particularly in high-traffic or remote areas where continuous surveillance is essential. Early attempts at automation used classical computer vision algorithms based on edge detection, motion tracking, or background subtraction. However, these approaches struggled with dynamic environments and were prone to false positives or missed detections, especially when dealing with irregular or occluded objects.

In recent years, deep learning-based object detection models—particularly the YOLO (You Only Look Once) family—have revolutionized the field by offering high-speed, real-time detection capabilities. Models such as YOLOv3 and YOLOv4 marked a leap forward in accuracy and generalization, but still faced challenges in detecting small, low-contrast, or partially obscured foreign objects, which are common in railway environments. YOLOv5 further improved performance with features like auto-learning bounding box anchors, mosaic data augmentation, and efficient model scaling. However, the standard YOLOv5s variant (the "small" model) still exhibits limitations when deployed on edge devices with constrained computational resources. It also struggles with the detection of small-scale intrusions such as wires, rocks, or debris, which may be critical threats. Furthermore, the reliance on predefined anchor boxes and fixed-scale feature extraction can negatively impact convergence speed, detection precision, and adaptability to complex backgrounds.

#### 1.4.1 EXISTINGSYSTEM DISADVANTAGES:

- False Positives and Missed Detections
- Dependence on High-Quality Training Data
- Limited Performance in Harsh Weather

### 1.5 LITERATURE SURVEY

**Title:** Railway obstacle intrusion warning mechanism integrating YOLO-based detection and risk assessment

**Author:** Z. Zhang, P. Chen, Y. Huang, L. Dai, F. Xu, and H. Hu

**Year:** 2024

**Description:** This paper presents a railway obstacle intrusion warning system that integrates YOLO-based object detection with a risk assessment module. The system aims to identify foreign objects on tracks in real-time, evaluate the associated risk level, and generate appropriate warnings. By utilizing YOLO for object detection, the proposed approach enhances detection accuracy and processing speed, making it more suitable for deployment in dynamic railway environments where safety is paramount.

**Title:** Railway intrusion detection based on machine vision: A survey, challenges, and perspectives

**Author**: Z. Cao, Y. Qin, L. Jia, Z. Xie, Y. Gao, Y. Wang, P. Li, and Z. Yu

**Year:** 2024

**Description:** This comprehensive survey reviews the progress in machine vision-based railway intrusion detection. The paper outlines the various deep learning models applied in the domain, discusses their strengths and limitations, and highlights real-world challenges such as occlusions, environmental variations, and small object detection. It also identifies future research directions, including lightweight models and improved sensor fusion techniques.

**Title:** EBSE-YOLO: High precision recognition algorithm for small target foreign object detection

**Author:** S. Wang, Y. Wang, Y. Chang, R. Zhao, and Y. She

**Year:** 2023

**Description:** EBSE-YOLO introduces an enhanced YOLO-based model aimed at improving the precision of small object detection, particularly for railway foreign objects. The model integrates edge enhancement and spatial encoding techniques to better identify and localize hard-to-detect small targets. Experimental results demonstrate that EBSE-YOLO significantly outperforms existing YOLO variants in detecting small and irregular objects under complex conditions.

**Title:** A novel strategy of two-stage cascaded CNN and overhaul knowledge distillation for fast railway foreign objects intrusion detection

**Author:** S. Meng, W. Chen, and Y. Jiang

**Year:** 2024

**Description:** This study proposes a two-stage detection system combining a cascaded CNN architecture with an Overhaul Knowledge Distillation framework to optimize detection speed and accuracy. The initial CNN quickly filters images with potential foreign objects, while the refined stage focuses on precise localization. The distillation technique helps train a lightweight model capable of maintaining high accuracy, making the approach suitable for real-time railway monitoring.

**Title:** An improved YOLOv5 method for small object detection in UAV capture scenes

**Author:** Z. Liu, X. Gao, Y. Wan, J. Wang, and H. Lyu

**Year:** 2023

**Description:** The paper proposes enhancements to the YOLOv5 architecture for detecting small objects in aerial images captured by UAVs, which share similar detection challenges as railway scenarios. Improvements include feature pyramid modifications and anchor optimization, resulting in better detection of small targets. The approach has been validated on several public UAV datasets and shows promise for applications in railway safety and infrastructure monitoring.

## 1.6 PROPOSED SYSTEM

The proposed system introduces a two-stage architecture designed to efficiently and accurately detect foreign objects on railway tracks. In the first stage, a lightweight image classification network acts as a fast filter, rapidly identifying whether a given railway image contains a foreign object. This stage significantly reduces the number of images passed to the heavier detection model, thereby enhancing the system's overall processing speed. To maintain high accuracy while keeping the classifier lightweight, the Overhaul Knowledge Distillation (OKD) method is employed. In this setup, a robust teacher network guides the training of the lightweight classifier, enabling it to learn high-quality feature representations and make precise decisions with minimal computational overhead.

Once an image is flagged as containing a foreign object, it is forwarded to the second stage, where the YOLOv8 object detection model is applied. YOLOv8 offers substantial improvements over earlier versions in terms of both accuracy and inference speed, making it ideal for real-time railway monitoring. This model precisely localizes and classifies the foreign objects, enabling timely responses to potential threats. By integrating YOLOv8 with a pre-filtering classification stage, the proposed system strikes an effective balance between speed and accuracy. The combined use of OKD and YOLOv8 ensures that the system not only reduces false positives and processing delays but also delivers state-of-the-art performance in real-world railway safety applications.

### 1.6.1 PROPOSED SYSTEM ADVANTAGES:

- High Accuracy
- Real-Time Performance
- Lightweight and Deployable
- Efficient Two-Stage System

# CHAPTER 2

# PROJECT DESCRIPTION

## 2.1 GENERAL:

The detection of foreign objects on railway tracks is crucial for ensuring the safety and smooth operation of train systems. Existing detection methods often suffer from low efficiency and suboptimal accuracy, which can lead to delayed responses and increased risk of accidents. To address these issues, this work introduces an enhanced railway foreign object intrusion detection framework that combines the power of YOLOv8, a state-of-the-art object detection algorithm, and Overhaul Knowledge Distillation, a technique for improving model efficiency.

The proposed system utilizes a two-stage architecture to enhance detection speed and accuracy. In the first stage, a lightweight image classification network quickly determines whether a railway image contains foreign objects, minimizing the need for computationally expensive object detection algorithms. This improves the overall speed of the system, allowing for faster decision-making. In the second stage, when foreign objects are flagged by the classification network, YOLOv8 is employed to detect and localize these objects precisely. YOLOv8 provides significant improvements in both detection accuracy and inference speed compared to earlier versions like YOLOv3, ensuring real-time performance in dynamic environments.

Furthermore, the framework incorporates Overhaul Knowledge Distillation to train the lightweight classification network. This method involves training the smaller model with the guidance of a larger, more robust network, helping it achieve high accuracy while maintaining low computational demands. The combination of these advanced techniques results in a system that offers high detection accuracy, improved processing speed, and better robustness in diverse environmental conditions. Experimental results demonstrate that the proposed method outperforms traditional approaches, providing state-of-the-art performance in terms of FPS (frames per second) and detection reliability.

## 2.2 METHODOLOGIES

### 2.2.1MODULES NAME:

**Modules Name:**

- Data Collection
- Annotations
- Yolov8 Model Building
- Feature Engineering
- Model Training
- Model Evaluation
- Object Detection

### 2.2.2 MODULES EXPLANATION:

**1) Data Collection:**

The first step in building the foreign object detection system is to gather a diverse and comprehensive dataset of railway images. These images should cover a variety of environmental conditions such as different lighting, weather, and track types. The dataset should include both "normal" images (no foreign objects present) and "intruded" images (with foreign objects on the tracks). Data can be collected from railway surveillance cameras, public datasets, or simulated environments. The diversity in the dataset ensures that the model is trained to generalize well across different real-world scenarios.

**2) Annotations:**

Once the data is collected, the next step is to annotate the images. In this stage, each image is labeled as either "normal" or "intruded" based on the presence of foreign objects. For the "intruded" images, further annotations are made to identify the specific type and location of the foreign objects within the image. This could involve drawing bounding boxes around the foreign objects and labeling them accordingly. The accuracy and quality of these annotations are crucial, as they directly impact the performance of the trained model.

**3) YOLOv8 Model Building:**

With the annotated dataset ready, the next step is to build the YOLOv8 model for object detection. YOLOv8 is an advanced deep learning model designed for real-time object detection. It is known for its high speed and accuracy, making it ideal for dynamic environments like railway tracks where timely detection is essential. The YOLOv8 model is built and configured to detect and localize foreign objects in the images. This involves selecting the appropriate architecture, defining the input and output layers, and configuring hyperparameters for optimal performance.

**4) Feature Engineering:**

Feature engineering involves identifying and selecting the most relevant features from the data that will help the model make accurate predictions. For object detection, this includes preparing the input images, ensuring they are resized and normalized, and possibly applying data augmentation techniques such as rotation, flipping, and zooming to create a more robust model. Additionally, features like texture, shape, and color patterns that are relevant to foreign object detection can be considered for enhancement.

**5) Model Training:**

The next step is to train the YOLOv8 model using the annotated dataset. During training, the model learns to identify patterns in the images that correspond to foreign objects and how to classify and localize them accurately. The training process includes backpropagation and optimization algorithms to minimize the loss function, iterating through many epochs to ensure that the model learns the relationships between the input data and the target outputs (object classification and localization). The Overhaul Knowledge Distillation technique is applied here to improve the efficiency and accuracy of the lightweight ResNet-tiny classification network, which is trained under the guidance of a larger, more robust network.

**6) Model Evaluation:**

After training, the model undergoes rigorous evaluation using a separate validation or test dataset. The goal is to assess how well the trained model generalizes to new, unseen images. Metrics such as accuracy, precision, recall, and F1-score are used to evaluate the classification performance, while mean average precision (mAP) is typically used for object detection tasks. Evaluation metrics like FPS (frames per second) and inference speed are also crucial to ensure that the model performs efficiently in real-time scenarios.

**7) Object Detection:**

Once the model has been successfully trained and evaluated, it is used for object detection on new images. In this final step, the trained YOLOv8 model takes in new railway images and identifies whether they contain foreign objects. For images classified as "intruded," the model not only classifies the object but also localizes it by drawing bounding boxes and categorizing the type of foreign object detected. The model is expected to perform in real-time, ensuring that it can continuously monitor and provide accurate results for immediate action on railway tracks. This process, from data collection to object detection, ensures that the proposed system is both accurate and efficient in detecting foreign objects on railway tracks, ultimately improving safety and operational efficiency.

## 2.3 TECHNIQUE USED OR ALGORITHM USED

### 2.3.1 EXISTING TECHNIQUE: -

**Yolov5**

The core algorithm utilized is YOLOv5s, a state-of-the-art single-stage object detector designed for real-time detection tasks. YOLO (You Only Look Once) divides the input image into grids and simultaneously predicts bounding boxes and class probabilities, providing high-speed inference. YOLOv5s is a lightweight variant optimized for faster processing with fewer parameters.

To improve feature extraction, the model integrates EfficientNet as a backbone network, which scales network depth, width, and resolution efficiently, achieving better performance with fewer computations. Furthermore, the Convolutional Block Attention Module (CBAM), embedded in the C3 module, enhances the model's focus on relevant spatial and channel-wise features, particularly benefiting small object detection.

The K-means++ clustering algorithm is employed to optimize anchor box dimensions (a priori frames), leading to better bounding box proposals that match the data distribution, improving training stability and detection accuracy.

### 2.3.2 PROPOSED TECHNIQUE USED OR ALGORITHM USED:

**Yolov8** This work uses YOLOv8 for accurate and real-time detection of foreign objects on railway tracks. Unlike earlier versions, YOLOv8 adopts an anchor-free design, making it better at detecting small and irregularly shaped objects. It also includes an improved backbone and head structure, helping the model learn features more effectively.

YOLOv8 offers higher speed and accuracy compared to YOLOv5. It supports better data augmentation, faster training, and improved detection performance in complex environments. Its lightweight design also makes it suitable for deployment on real-time surveillance systems.

In the proposed system, YOLOv8 is used as the second stage after a lightweight classifier filters the input images. This two-step approach reduces the workload on the detection model while maintaining high accuracy, making the system efficient and reliable for large-scale railway monitoring.

## CHAPTER 3

## REQUIREMENTS ENGINEERING

### 3.1 GENERAL

We can see from the results that on each database, the error rates are very low due to the discriminatory power of features and the regression capabilities of classifiers. Comparing the highest accuracies (corresponding to the lowest error rates) to those of previous works, our results are very competitive.

### 3.2 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

- PROCESSOR : DUAL CORE 2 DUOS.
- RAM : 4GB DD RAM
- HARD DISK : 250 GB

### 3.3 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

Operating System : Windows 10/11

Platform : Spyder3

Programming Language : Python

Front End : Spyder3

### 3.4 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior, Firstly, the system is the first that achieves the standard notion of semantic security for data confidentiality in attribute-based deduplication systems by resorting to the hybrid cloud architecture.

### 3.5 NON-FUNCTIONAL REQUIREMENTS

**The major non-functional Requirements of the system are as follows**

**Usability**

The system is designed with completely automated process hence there is no or less user intervention.

**Reliability**

The system is more reliable because of the qualities that are inherited from the chosen platform python. The code built by using python is more reliable.

**Performance**

This system is developing in the high level languages and using the advanced back-end technologies it will give response to the end user on client system with in very less time.

**Supportability**

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is built into the system.

**Implementation**

The system is implemented in web environment using Jupyter notebook software. The server is used as the intellignce server and windows 10 professional is used as the platform. Interface the user interface is based on Jupyter notebook provides server system.

## CHAPTER 4

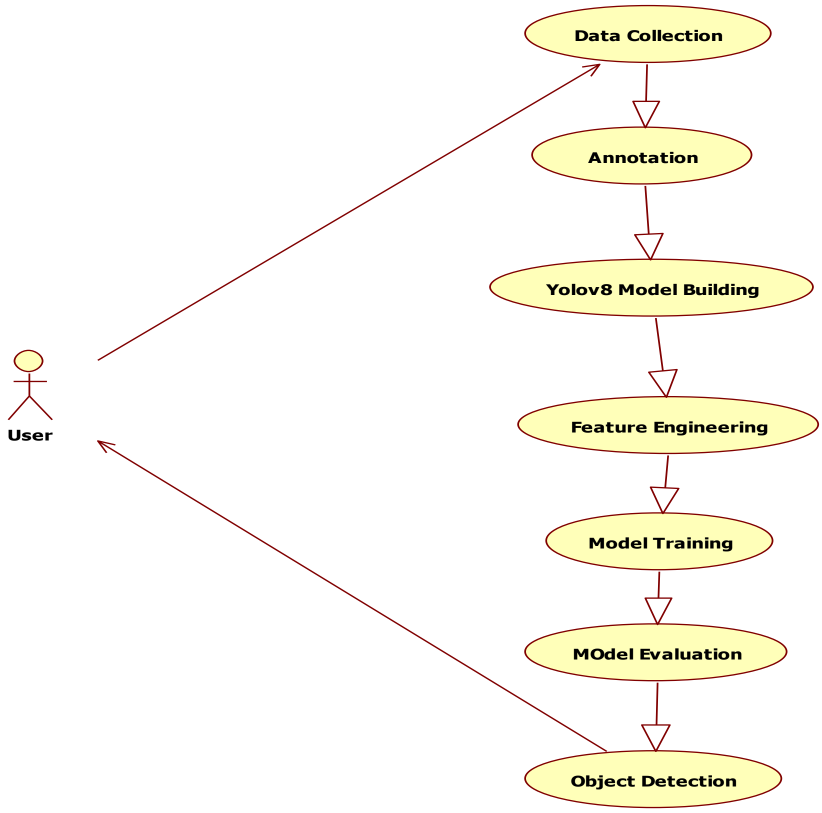
## DESIGN ENGINEERING

### 4.1 GENERAL

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering.

## 4.2 UML DIAGRAMS

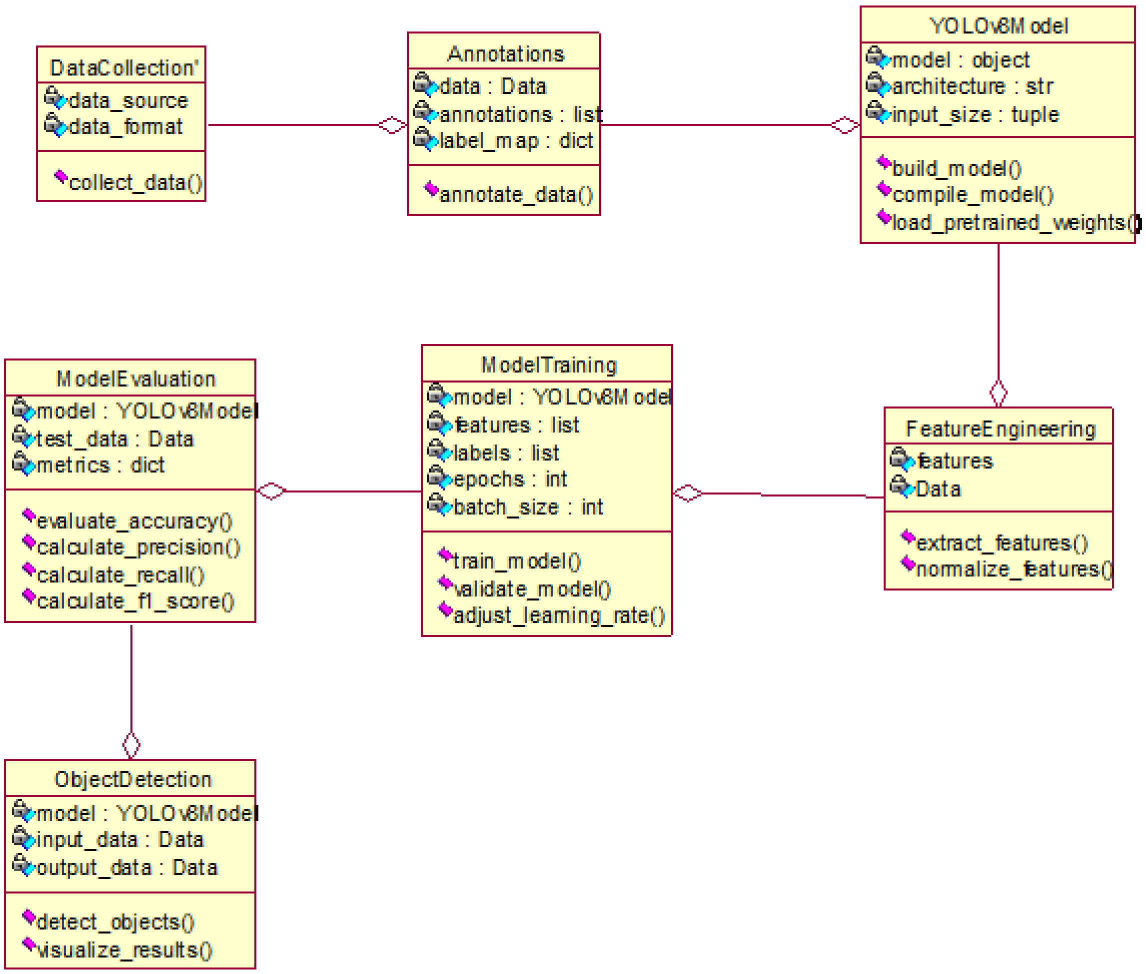
### 4.2.1 USE CASE DIAGRAM




**EXPLANATION:**

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The above diagram consists of user as actor. Each will play a certain role to achieve the concept.

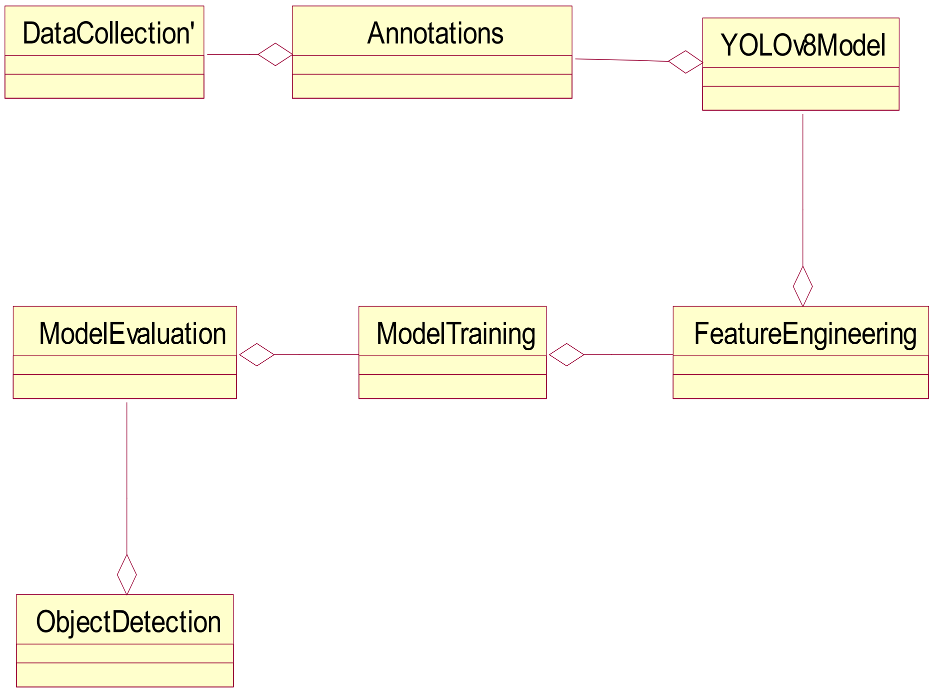
### 4.2.2 CLASS DIAGRAM




### EXPLANATION

In this class diagram represents how the classes with attributes and methods are linked together to perform the verification with security. From the above diagram shown the various classes involved in our project.

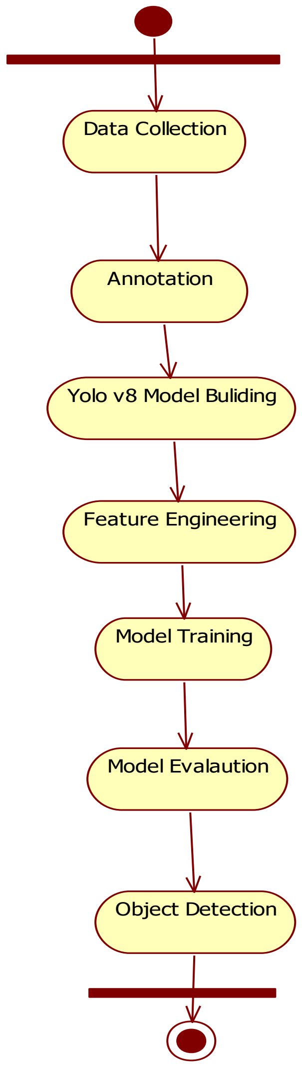
### 4.2.3 OBJECT DIAGRAM




**EXPLANATION:**

In the above digram tells about the flow of objects between the classes. It is a diagram that shows a complete or partial view of the structure of a modeled system. In this object diagram represents how the classes with attributes and methods are linked together to perform the verification with security.

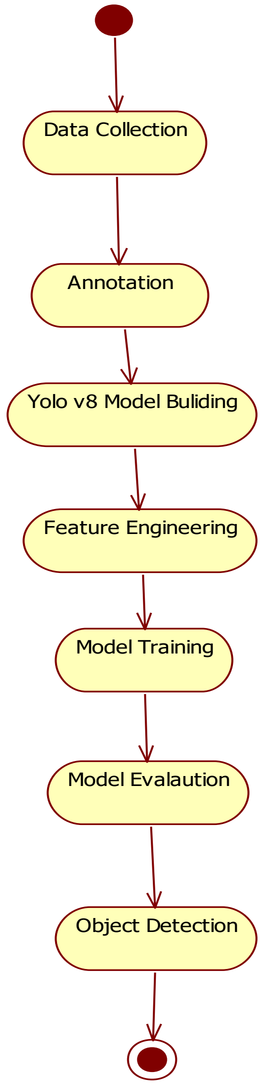
### 4.2.4 STATE DIAGRAM




**EXPLANATION:**

State diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

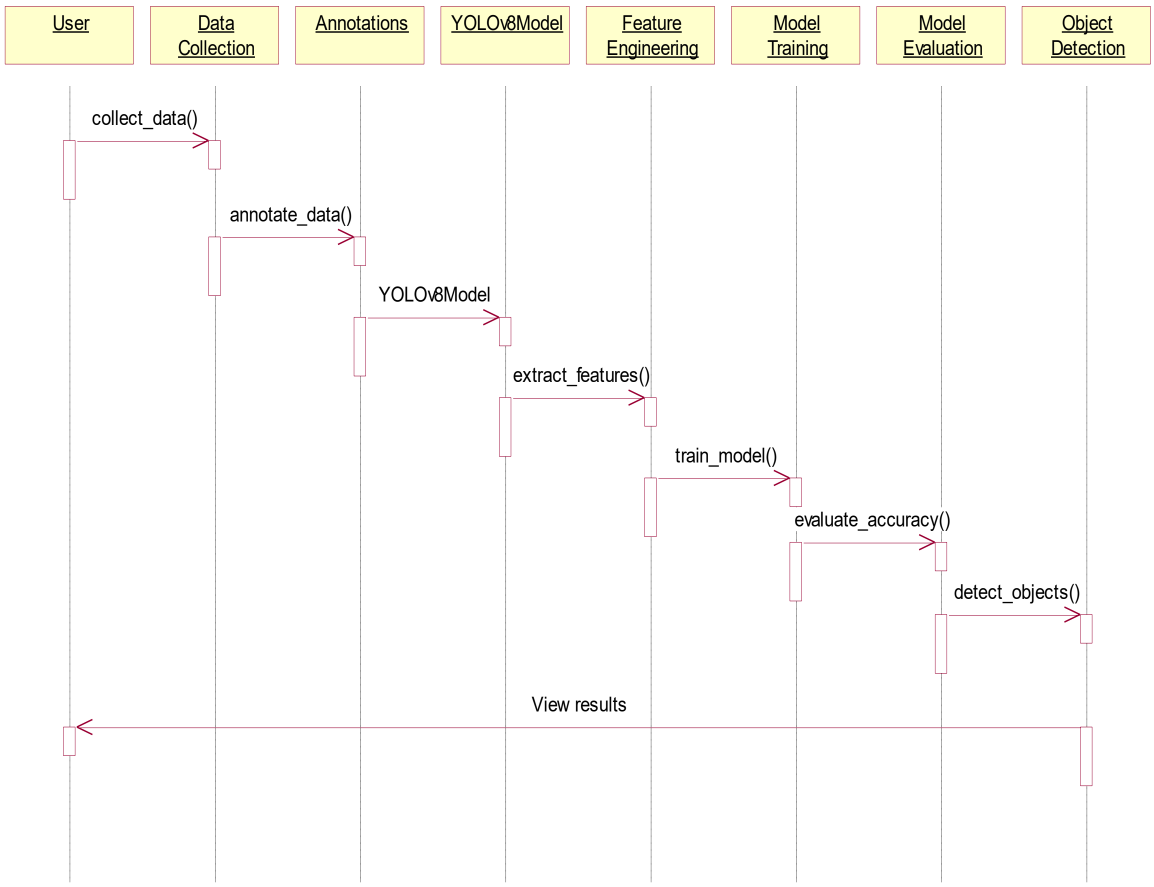
### 4.2.5 ACTIVITY DIAGRAM




**EXPLANATION:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

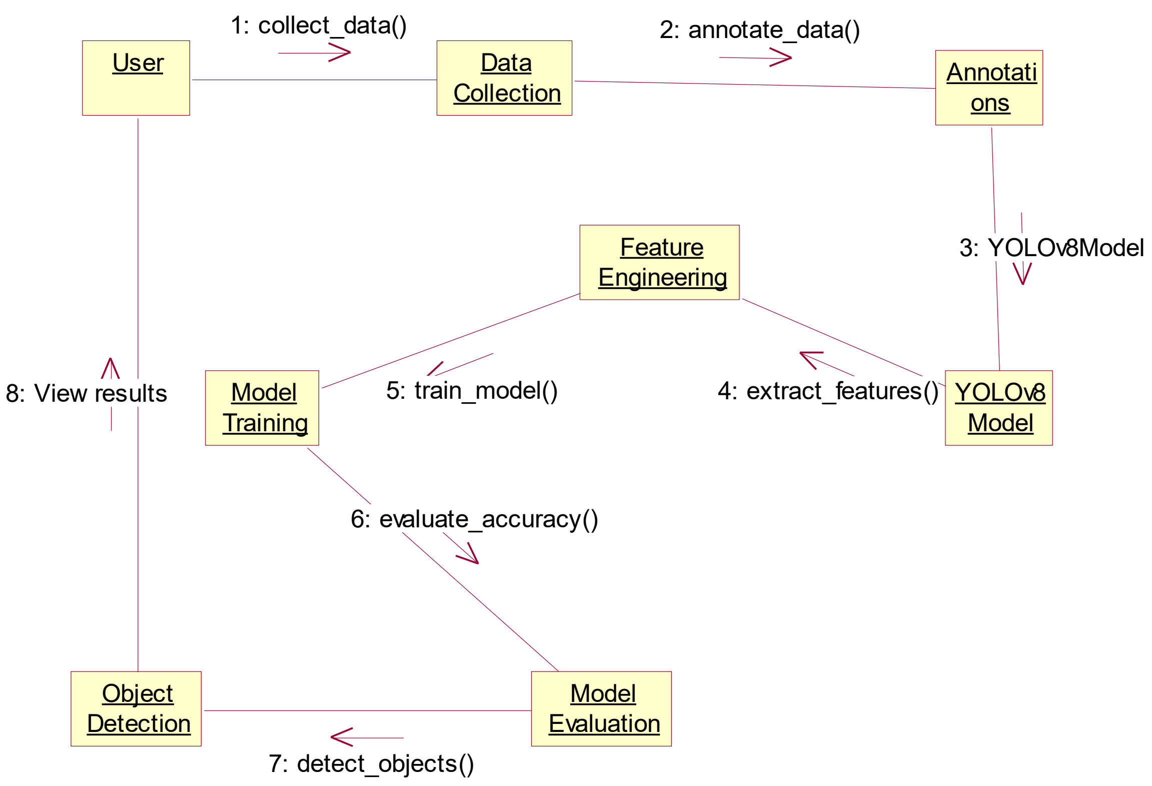
### 4.2.6 SEQUENCE DIAGRAM




**EXPLANATION:**

A sequence diagram in Unified Modeling Language (UML) 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.

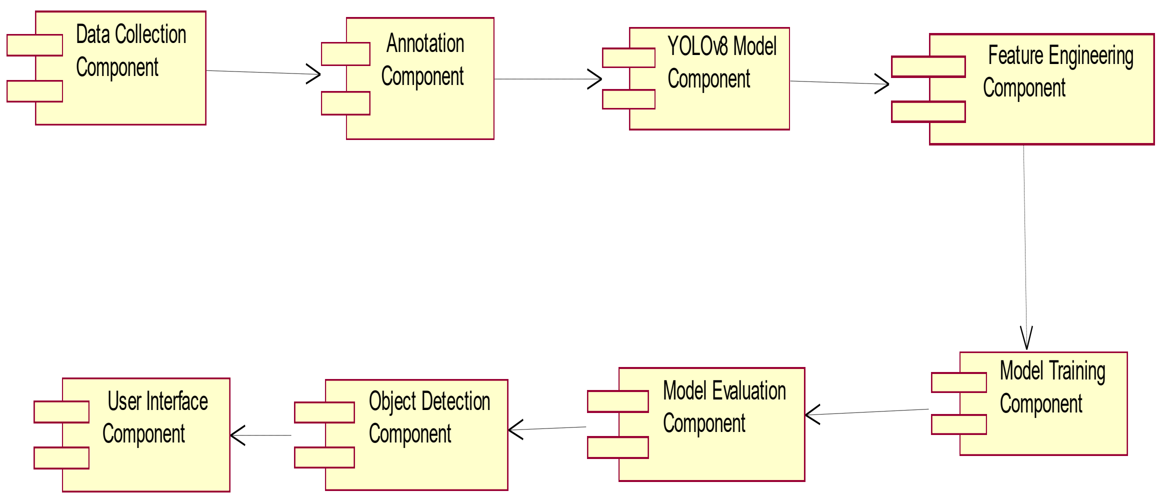
### 4.2.7 COLLABORATION DIAGRAM




### EXPLANATION:

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

### 4.2.8 COMPONENT DIAGRAM




### EXPLANATION

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

### 4.2.9 DATA FLOW DIAGRAM

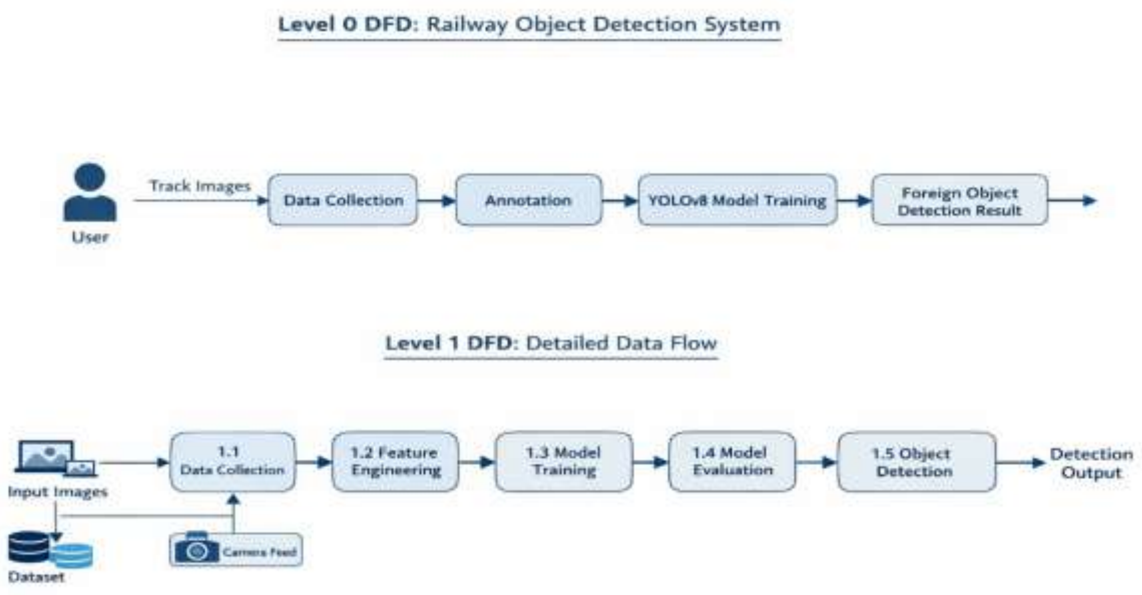


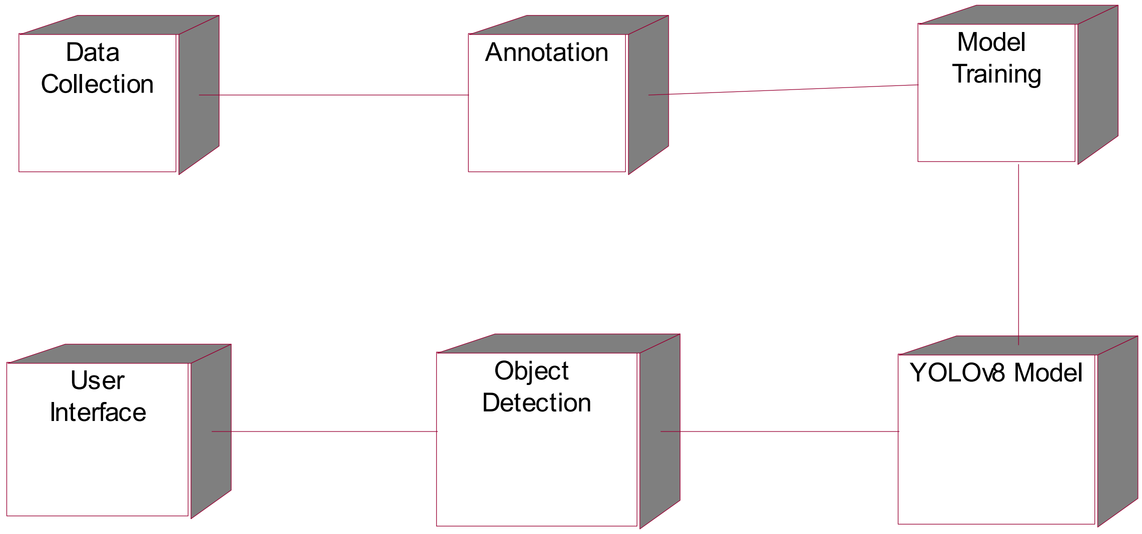

Fig 4.9: Data Flow Diagrams

**EXPLANATION:**

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kinds of data will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

### 4.2.10 DEPLOYMENT DIAGRAM




**EXPLANATION:**

Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. It maps software pieces of a system to the device that are going to execute it.

**SYSTEM ARCHITECTURE:**

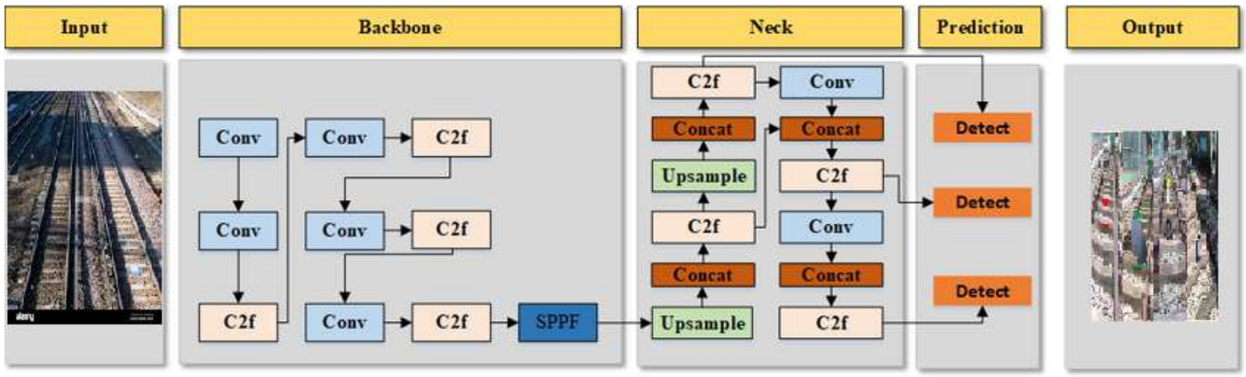



Fig 4.11: System Architecture

## CHAPTER 5

## DEVELOPMENT TOOLS

### 5.1 Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

### 5.2 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, Small Talk, 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.

### 5.3 Importance of Python

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

### 5.4 Features of Python

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.

- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

**5.5 Libraries used in python**

- numpy - mainly useful for its N-dimensional array objects.
- pandas - Python data analysis library, including structures such as dataframes.
- matplotlib - 2D plotting library producing publication quality figures.
- scikit-learn - the machine learning algorithms used for data analysis and data mining tasks.


Figure : NumPy, Pandas, Matplotlib, Scikit-learn

## CHAPTER 6

## SOFTWARE TESTING

### 8.1 GENERAL

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### 8.2 DEVELOPING METHODOLOGIES

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

### 8.3Types of Tests

#### 8.3.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

#### 8.3.2 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.
Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.
Systems/Procedures: interfacing systems or procedures must be invoked.

**8.3.3 System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**8.3.4 Performance Test**

The Performance test ensures that the output be produced within the time limits,and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

**8.3.5 Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**8.3.6 Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Acceptance testing for Data Synchronization:**

- The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
- The Route add operation is done only when there is a Route request in need
- The Status of Nodes information is done automatically in the Cache Updation process

**8.2.7 Build the test plan**

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

**CHAPTER 7**

**FUTURE ENHANCEMENT**

**9.1 FUTURE ENHANCEMENTS:**

An impactful future enhancement for the proposed railway foreign object intrusion detection system would be the integration of a self-supervised learning framework to enable the model to continuously improve from real-world, unlabeled data. Unlike traditional supervised approaches, self-supervised methods can leverage the vast amounts of video and sensor data collected by surveillance systems without the need for manual annotations. This continuous learning capability would allow the model to adapt over time to changing environmental conditions, emerging obstacle types, and evolving visual patterns on the railway tracks.

Furthermore, implementing a feedback-based active learning loop—where difficult or ambiguous detection instances are flagged and sent for human review—can enrich the training dataset and gradually improve model performance. This human-in-the-loop strategy would not only reduce false positives and false negatives but also help the system stay up-to-date with rare and novel intrusion events. Combined with edge computing for on-site processing and fast response, this approach could significantly enhance the system's adaptability, accuracy, and reliability in dynamic, real-world railway environments.

## CHAPTER 8

## CONCLUSIONAND REFERENCES

### 10.1 CONCLUSION

The proposed rapid railway foreign object intrusion detection system presents a significant advancement in enhancing rail safety through the application of modern deep learning techniques. By integrating a two-stage architecture that combines a lightweight classification model with the powerful YOLOv8 object detector, the system achieves high detection accuracy while maintaining computational efficiency—making it suitable for real-time deployment in dynamic railway environments. Furthermore, the incorporation of Overhaul Knowledge Distillation enables the lightweight model to retain high performance with reduced resource requirements, addressing the challenges of real-time edge deployment. This framework not only reduces the risk of accidents caused by track obstructions but also minimizes the need for manual inspections, thus improving operational efficiency. The system lays a strong foundation for future improvements and broader implementation across railway networks, contributing meaningfully to the modernization and safety of transportation infrastructure.

### 10.2 REFERENCES

[1] Z. Zhang, P. Chen, Y. Huang, L. Dai, F. Xu, and H. Hu, ''Railway obstacle intrusion warning mechanism integrating YOLO-based detection and risk assessment,'' J. Ind. Inf. Integr., vol. 38, Mar. 2024, Art. no. 100571.
[2] M. Givoni, ''Development and impact of the modern high-speed train:A review,'' Transp. Rev., vol. 26, no. 5, pp. 593–611, Sep. 2006.[3] Z. Cao, Y. Qin, L. Jia, Z. Xie, Y. Gao, Y. Wang, P. Li, and Z. Yu, ''Railway intrusion detection based on machine vision: A survey, challenges,
and perspectives,'' IEEE Trans. Intell. Transp. Syst., vol. 25, no. 7,pp. 6427–6448, Jul. 2024, doi: 10.1109/TITS.2024.3412170.
[4] T. Wang, Z. Zhang, and K.-L. Tsui, ''A deep generative approach for rail foreign object detections via semisupervised learning,'' IEEE Trans. Ind. Informat., vol. 19, no. 1, pp. 459–468, Jan. 2023.
[5] M. Simonović, M. Banić, D. Stamenković, M. Franke, K. Michels, I. Schoolmann, D. Ristic-Durrant, C. Ulianov, S. Stan, A. Plea, and M. Dimec, ''Toward the enhancement of rail sustainability: Demonstration of a holistic approach to obstacle detection in operational railway
environments,'' in Proc. Sustainability, vol. 16, Mar. 2024, p. 2613.
[6] T. Shi, P. Guo, R. Wang, Z. Ma, W. Zhang, W. Li, H. Fu, and H. Hu, ''A survey on multi-sensor fusion perimeter intrusion detection in highspeed railways,'' Sensors, vol. 24, no. 17, p. 5463, Aug. 2024.
[7] A. A. Aziz, K. A. Mohamad, and A. Alias, ''Obstacle detection system for railways using IoT sensors,'' Evol. Electr. Electron. Eng., vol. 1, no. 1, pp. 57–63, Dec. 2020.
[8] M. Fan, S. Tian, K. Liu, J. Zhao, and Y. Li, ''Infrared small target detection based on region proposal and CNN classifier,'' Signal, Image Video Process., vol. 15, no. 8, pp. 1927–1936, Nov. 2021.
[9] M. Rodriguez, S. Ali, and T. Kanade, ''Tracking in unstructured crowded scenes,'' in Proc. IEEE 12th Int. Conf. Comput. Vis., Sep. 2009, pp. 1389–1396.
[10] Z. Silar and M. Dobrovolny, ''Utilization of directional properties of optical flow for railway crossing occupancy monitoring,'' in Proc. Int. Conf. IT Converg. Secur. (ICITCS), Dec. 2013, pp. 1–4.
[11] C. Tastimur, M. Karakose, and E. Akin, ''Image processing based level crossing detection and foreign objects recognition approach in railways,'' Int. J. Appl. Math., Electron. Comput., vol. 1, pp. 19–23, Aug. 2017.
[12] M. Shah, O. Javed, and K. Shafique, ''Automated visual surveillance in realistic scenarios,'' IEEE MultimediaMag., vol. 14, no. 1, pp. 30–39, Jan. 2007. [13] Y. Sheikh, Y. Zhai, K. Shafique, and M. Shah, ''Visual monitoring of railroad grade crossing,'' Proc. SPIE, vol. 5403, 654, Sep. 2004.

[14] Y. Xiao, Z. Tian, J. Yu, Y. Zhang, S. Liu, S. Du, and X. Lan, ''A review of object detection based on deep learning,'' Multimedia Tools Appl., vol. 79, pp. 23729–23791, Jun. 2020.
[15] H. Alawad, S. Kaewunruen, and M. An, ''A deep learning approach towards railway safety risk assessment,'' IEEE Access, vol. 8, pp. 102811–102832, 2020.
[16] T. Ye, Z. Zhang, X. Zhang, and F. Zhou, ''Autonomous railway traffic object detection using feature-enhanced single-shot detector,'' IEEE Access, vol. 8, pp. 145182–145193, 2020.
[17] S. Tang and Y. Yuan, ''Object detection based on convolutional neural network,'' in Proc. Int. Conf., Jan. 2020, pp. 1–8.
[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, ''SSD: Single shot MultiBox detector,'' in Proc. 14th Eur. Conf. Comput. Vision, Amsterdam, The Netherlands. Springer, Oct. 2016, pp. 21–37.
[19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, ''Rich feature hierarchies for accurate object detection and semantic segmentation,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 580–587.
[20] R. Girshick, ''Fast R-CNN,'' in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1440–1448.
[21] J. Dai, Y. Li, K. He, and J. Sun, ''R-FCN: Object detection via regionbased fully convolutional networks,'' in Proc. Adv. Neural Inf. Process. Syst., Jan. 2016, pp. 1–13.
[22] S. Ren, K. He, R. Girshick, and J. Sun, ''Faster R-CNN: Towards realtime object detection with region proposal networks,'' IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[23] Z. Cong and X. Li, ''Track obstacle detection algorithm based on YOLOv3,'' in Proc. 13th Int. Congr. Image Signal Process., Biomed. Eng. Informat. (CISP-BMEI), Oct. 2020, pp. 12–17. [24] Z. Liu, X. Gao, Y. Wan, J. Wang, and H. Lyu, ''An improved YOLOv5 method for small object detection in UAV capture scenes,'' IEEE Access, vol. 11, pp. 14365–14374, 2023.
[25] Z. Li, Z. Rao, L. Ding, B. Ding, J. Fang, and X. Ma, ''YOLOv5s-D: A railway catenary dropper state identification and small defect detection model,'' Appl. Sci., vol. 13, no. 13, p. 7881, Jul. 2023.
[26] S. Wang, Y. Wang, Y. Chang, R. Zhao, and Y. She, ''EBSE-YOLO: High precision recognition algorithm for small target foreign object detection,'' IEEE Access, vol. 11, pp. 57951–57964, 2023.
[27] H. Li, Y. Dong, Y. Liu, and J. Ai, ''Design and implementation of UAVs for Bird's nest inspection on transmission lines based on deep learning,'' Drones, vol. 6, no. 9, p. 252, Sep. 2022.
[28] S. Ning, F. Ding, and B. Chen, ''Research on the method of foreign object detection for railway tracks based on deep learning,'' Sensors, vol. 24, no. 14, p. 4483, Jul. 2024.
[29] S. Meng, W. Chen, and Y. Jiang, ''A novel strategy of two-stage cascaded CNN and overhaul knowledge distillation for fast railway foreign objects intrusion detection,'' IEEE Access, vol. 12, pp. 127825–127835, 2024.
[30] S. Bhatnagar, L. Gill, S. Regan, S. Waldren, and B. Ghosh, ''A nested drone-satellite approach to monitoring the ecological conditions of wetlands,'' ISPRS J. Photogramm. Remote Sens., vol. 174, pp. 151–165, Apr. 2021.
[31] F. Kleinschroth, K. Banda, H. Zimba, S. Dondeyne, I. Nyambe, S. Spratley, and R. S. Winton, ''Drone imagery to create a common understanding of landscapes,'' Landscape Urban Planning, vol. 228, Dec. 2022, Art. no. 104571.
[32] P. Goodrich, O. Betancourt, A. C. Arias, and T. Zohdi, ''Placement and drone flight path mapping of agricultural soil sensors using machine learning,'' Comput. Electron. Agricult., vol. 205, Feb. 2023, Art. no. 107591.
[33] U. Iqbal, M. Z. B. Riaz, J. Zhao, J. Barthelemy, and P. Perez, ''Drones for flood monitoring, mapping and detection: A bibliometric review,'' Drones, vol. 7, no. 1, p. 32, Jan. 2023.
[34] İ. Bekmezci, O. K. Sahingoz, and Ş. Temel, ''Flying Ad-hoc networks (FANETs): A survey,'' Ad Hoc Netw., vol. 11, no. 3, pp. 1254–1270, May 2013.
[35] A. Puri, ''A survey of unmanned aerial vehicles (UAV) for traffic surveillance,'' Dept. of Computer Science and Engineering, Univ. South Florida, Tampa, FL, USA, Tech. Rep. 4202, 2005, pp. 1–29.
[36] M. Tan and Q. Le, ''Efficientnet: Rethinking model scaling for convolutional neural networks,'' in Proc. Int. Conf. Mach. Learn., 2019, pp. 6105–6114.
[37] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, ''Cbam: Convolutional block attention module,'' in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 3–19.
[38] D. Arthur and S. Vassilvitskii, ''K-means++: The advantages of careful seeding,'' in Proc. Soda,Jan.2007,pp.1027–1035.