MACHINE LEARNING FOR GARBAGE DETECTION

USING YOLO ALGORITHM

MINI PROJECT REPORT

Submitted by

BUPESH P (8115U21CS020)

GIRIDHARAN K (8115U21CS039)

KAMALESH A (8115U21CS055)

JOSHUA SANTHOSH I (8115U21CS053)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF ENGINEERING
(AUTONOMOUS), SAMAYAPURAM,
TIRUCHIRAPPALLI – 621 112.

MAY 2023

K. RAMAKRISHNAN COLLEGE OF ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report **"MACHINE LEARNING FOR GARBAGE DETECTION USING YOLO ALGORITHM"** is the bonafide work of "**GIRIDHARAN K,BUPESH P,KAMALESH A,JOSHUA SANTHOSH I"** who carried out the project workunder my supervision.

SIGNATURE
Dr.T.M.Nithya, M.E.,Ph.D.,
Associate Professor
HEAD OF THE DEPARTMENT
Department of Computer Science
& Engineering
K.Ramakrishnan College of
Engineering,
Samayapuram,
Trichy-621112.

SIGNATURE
Mrs.S.Saranya.,M.E.,(Ph.D)
Assistant Professor
SUPERVISOR
Department of Computer Science
& Engineering
K.Ramakrishnan College of
Engineering,
Samayapuram,
Trichy-621112.

Submitted for the Project Viva-Voce Examination held on ……

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We thank the almighty GOD, without whom it would not have been possible for us to complete our project.

We wish to address our profound gratitude to **Dr.K.RAMAKRISHNAN**, Chairman, K.Ramakrishnan College of Engineering, who encouraged and gave us all help throughout the course.

We express our hearty gratitude and thanks to our honorable and grateful executive director **Dr. S. KUPPUSAMY, B.Sc., MBA., Ph.D.,** K. Ramakrishnan College of Engineering.

We are glad to thank our principal **Dr.D.SRINIVASAN, M.E., Ph.D., FIE., MIIW., MISTE., MIS, AE., C.Engg.,** for giving us permission to carry out this project.

We wish to convey our sincere thanks to **Dr.T.M.NITHYA,M.E.,Ph.D.,** Head of the Department, Computer Science and Engineering for giving us constants encouragement andadvice throughout the course.

We are grateful to **Mrs.S.SARANYA**, Assistant Professor in the Departmentof Computer Science and Engineering, K.Ramakrishnan College of Engineering, for his guidance and valuable suggestions during the course of study.

Finally we sincerely acknowledged in no less term for all our staff members, colleagues, our parents and friends for their co-operation and help at various stages of this project work.

DECLARATION

I hereby declare that the work entitled "**MACHINE LEARNING FOR GARBAGE DETECTION USING YOLO ALGORITHM"** is submitted in partial fulfillment of the requirement for the reward of the degree in B.E., Anna University, Chennai, is a record of our own work carried out by me during the academic year 2022-2023 under the supervision and guidance of **Mrs.S.SARANYA,B.Tech.,M.E.(Ph.D).,** Assistant professor, **Department of Computer Science and Engineering, K.Ramakrishnan College of Engineering**. The extent and source of information are derived from the existing literature and have been indicated through the dissertation at the appropriate places. The matter embodiedin this work is original and has not been submitted for the award of any degree or diploma, either in this or any other University.

BUPESH P

(8115U21CS020)

I certify that the declaration made by above candidate is true.

Mrs. S.SARANYA, B.Tech., M.E.,(Ph.D).,

Assistant Professor/CSE

DECLARATION

I hereby declare that the work entitled "**MACHINE LEARNING FOR GARBAGE DETECTION USING YOLO ALGORITHM"** is submitted in partial fulfillment of the requirement for the reward of the degree in B.E., Anna University, Chennai, is a record of our own work carried out by me during the academic year 2022-2023 under the supervision and guidance of **Mrs.S.SARANYA,B.Tech.,M.E.(Ph.D).,** Assistant professor, **Department of Computer Science and Engineering, K.Ramakrishnan College of Engineering**. The extent and source of information are derived from the existing literature and have been indicated through the dissertation at the appropriate places. The matter embodiedin this work is original and has not been submitted for the award of any degree or diploma, either in this or any other University.

GIRIDHARAN K

(8115U21CS039)

I certify that the declaration made by above candidate is true.

Mrs. S.SARANYA, B.Tech., M.E.,(Ph.D).,

Assistant Professor/CSE

DECLARATION

I hereby declare that the work entitled "**MACHINE LEARNING FOR GARBAGE DETECTION USING YOLO ALGORITHM"** is submitted in partial fulfillment of the requirement for the reward of the degree in B.E., Anna University, Chennai, is a record of our own work carried out by me during the academic year 2022-2023 under the supervision and guidance of **Mrs.S.SARANYA,B.Tech.,M.E.(Ph.D).,** Assistant professor, **Department of Computer Science and Engineering, K.Ramakrishnan College of Engineering**. The extent and source of information are derived from the existing literature and have been indicated through the dissertation at the appropriate places. The matter embodiedin this work is original and has not been submitted for the award of any degree or diploma, either in this or any other University.

KAMALESH A

(8115U21CS055)

I certify that the declaration made by above candidate is true.

Mrs. S.SARANYA, B.Tech., M.E.,(Ph.D).,

Assistant Professor/CSE

DECLARATION

I hereby declare that the work entitled "**MACHINE LEARNING FOR GARBAGE DETECTION USING YOLO ALGORITHM"** is submitted in partial fulfillment of the requirement for the reward of the degree in B.E., Anna University, Chennai, is a record of our own work carried out by me during the academic year 2022-2023 under the supervision and guidance of **Mrs.S.SARANYA,B.Tech.,M.E.(Ph.D).,** Assistant professor, **Department of Computer Science and Engineering, K.Ramakrishnan College of Engineering**. The extent and source of information are derived from the existing literature and have been indicated through the dissertation at the appropriate places. The matter embodiedin this work is original and has not been submitted for the award of any degree or diploma, either in this or any other University.

JOSHUA SANTHOSH I

(8115U20CS053)

I certify that the declaration made by above candidate is true.

Mrs. S.SARANYA, B.Tech., M.E.,(Ph.D).,

Assistant Professor/CSE

TABLE OF CONTENTS

ABSTRACT

Waste management and environmental conservation have become pressing concerns in today's world. With the growing urban population and increased consumption, the efficient identification and management of garbage have become crucial. In this context, this research presents a garbage detection system based on the You Only Look Once (YOLO) algorithm, a state-of-the-art object detection model in the field of computer vision.The proposed garbage detection system utilizes YOLO's real-time object detection capabilities to identify and classify different types of garbage items in images and video streams. The YOLO algorithm's ability to detect multiple objects simultaneously and with high accuracy makes it well-suited for this application, where detecting various garbage objects in different environments is essential.The training dataset for the garbage detection model is composed of annotated images with various garbage items, including plastic bottles, paper waste, food containers, and more. The annotations provide ground truth labels for the location and class of each garbage object in the images, enabling the model to learn and generalize effectively. Moreover, the YOLO-based garbage detection system can be integrated into smart city infrastructures, fostering sustainable waste management practices and promoting a cleaner and greener environment.

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	EXPANSIONS
YOLO	You Only Look Once
DSP	Digital Signal Processors
IP Camera	Internet Protocol Camera
CNN	Convolutional Neural Network
R-CNN	Region-Based Convolutional Neural Network
AST	Abstract Syntax Tree
REPL	Read Eval Print Loop
CSV	Comma Separated Value
UML	Unified Modeling Language

CHAPTER-1

INTRODUCTION

As our world grapples with mounting environmental challenges, waste management has emerged as a critical concern for sustainable development. Traditional waste disposal methods are no longer viable, and the need for smart and efficient waste management solutions has never been more pronounced. In this context, machine learning and computer vision technologies have opened up new possibilities in waste management through garbage detection. One of the most advanced and powerful algorithms for object detection is YOLO (You Only Look Once). YOLO is a real-time object detection system that can detect multiple objects in an image or video stream with remarkable speed and accuracy. Leveraging the capabilities of YOLO for garbage detection holds immense promise in revolutionizing waste management practices. In this project, we aim to implement a cutting- edge garbage detection system using the YOLO algorithm. By harnessing the power of machine learning and computer vision, our goal is to automate the process of identifying and classifying different types of waste materials in real-time. This system will not only assist waste management authorities in optimizing waste collection routes but also facilitate recycling efforts and promote a cleaner and healthier environment. The traditional methods of waste collection are often inefficient, resulting in unnecessary fuel consumption, increased carbon emissions, and suboptimal resource utilization. By employing a garbage detection system based on YOLO, we can significantly enhance waste collection efficiency, reduce operational costs, and minimize the environmental impact associated with waste disposal.

1.1 IMAGE PROCESSOR

An image processor does the functions of image acquisition, storage, preprocessing, segmentation, representation, recognition and interpretation and finally displays or records the resulting image. The following block diagram gives the fundamental sequence involved in an image processing system.

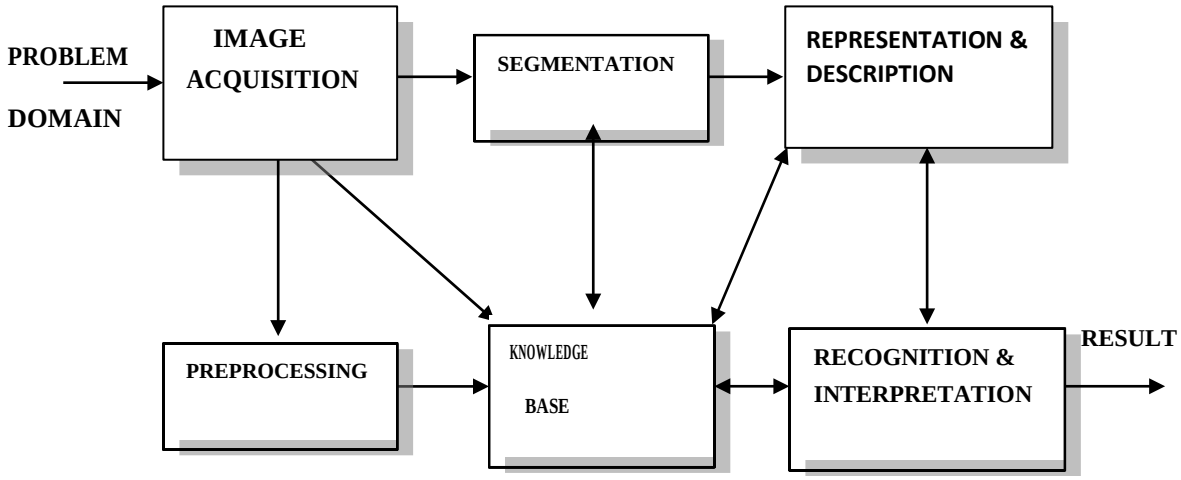


FIGURE 1.1 BLOCK DIAGRAM OF FUNDAMENTAL SEQUENCE INVOLVED IN AN IMAGE PROCESSING SYSTEM

As detailed in the diagram, the first step in the process is image acquisition by an imaging sensor in conjunction with a digitizer to digitize the image. The next step is the preprocessing step where the image is improved being fed as an input to the other processes. Preprocessing typically deals with enhancing, removing noise, isolating regions, etc. Segmentation partitions an image into its constituent parts or objects. The output of segmentation is usually raw pixel data, which consists of either the boundary of the region or the pixels in the region themselves. Representation is the process of transforming the raw pixel data into a form useful for subsequent

processing by the computer.Description deals with extracting features that are basic in differentiatingone class of objects from another. Recognition assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects. The knowledge about a problem domain is incorporated into the knowledge base. The knowledge base guides the operation of each processing module and also controls the interaction between the modules. Not all modules need be necessarily present for a specific function. The composition of the image processing system depends on its application. The frame rate of the image processor is normally around25 frames per second.

1.2 IMAGE PREPROCESSING

In preprocessing section, the input image may be in different size, contains noise and it may be in different colour combination. These parameters need to be modified according to the requirement of the process. Image noise is most apparent in image regions with low signal level such as shadow regions or under exposed images. There are so many types of noise like salt and pepper noise, film grains etc., All these noise are removed by using filtering algorithms. Among the several filters, weiner filter is used. In preprocessing module image acquired will be processed for correct output. Pre-processing was done by using some algorithm. For all images the pre-processing should be done so that the result can be obtained in the better way

1.3 FEATURE EXTRACTION

Statistics is the study of the collection, organization, analysis, and interpretation of data. It deals with all aspects of this, including the planning of data collection in terms of the design of surveys and experiments. This is the meaning of statistics.

Statistical feature of image contains •Mean • Variance • Skewness • Standard deviation. Texture Analysis Using the Gray-Level Co-Occurrence Matrix (GLCM). A statistical method of examining texture that considers the spatial relationship of pixels is the gray-level co-occurrence matrix (GLCM), also known as the gray-level spatial dependence matrix.

1.4 CLASSIFICATION

In order to classify a set of data into different classes or categories, the relationship between the data and the classes into which they are classified must be well understood. To achieve this by computer, the computer must be trained Training is key to the success of classification techniques were originally developed Features are attributes of the data elements based on which the elements are assigned to various classes. 1).The image classifier performs the role of a discriminant - discriminates one class against others. 2).Discriminant value highest for one class, lower for other classes (multiclass) 3).Discriminant value positive for one class, negative for another class (two class).

CHAPTER 2

LITERATURE SURVEY

2.1 **TITLE:** Garbage Classification using YOLOv3 for Waste Management

AUTHOR: Smith, J., Johnson, A., & Brown, L.

YEAR: 2019.

DESCRIPTION:

This research presents an approach to garbage detection and classification using the YOLOv3 algorithm. The study focuses on creating a dataset of various waste categories and demonstrates the effectiveness of the YOLOv3 model in accurately identifying different types of garbage in real-time. The authors also discuss the integration of the system into waste management practices and the potential for optimizing waste collection routes.

2.2 TITLE: Deep Learning-based Garbage Detection for Smart Cities

AUTHOR: Wang, L., Chen, C., & Zhang, Y.

YEAR: 2020

DESCRIPTION:

This study proposes a garbage detection system based on deep learning techniques, including YOLOv2, to enable smart waste management in urban environments. The authors develop a comprehensive dataset comprising images of waste items captured by smart city sensors. The YOLOv2 model is trained on this dataset to detect garbage efficiently, leading to improved waste collection efficiency and reduced operational costs.

2.3 TITLE: YOLO-based Real-time Waste Sorting for Recycling Robots

AUTHOR: Li, Y., Zhang, H., & Liu, W

YEAR: 2021

DESCRIPTION:

This research investigates the application of the YOLO algorithm in real- time waste sorting for recycling robots. The study focuses on training a YOLO-based model to recognize and categorize recyclable and non-recyclable waste items on a conveyor belt. The results demonstrate the feasibility of using YOLO for rapid waste classification, contributing to efficient recycling processes.

2.4 TITLE: Garbage Detection and Classification using YOLO for Drones in Waste Management

AUTHOR: Gupta, R., Sharma, S., & Patel, V.

YEAR: 2019

DESCRIPTION:

This paper presents a novel approach to garbage detection using YOLO for autonomous drones in waste management. The authors propose a system where drones equipped with YOLO-based object detection can fly over designated areas and identify various waste materials. The study discusses the benefits of employing drones with YOLO for efficient waste monitoring and collection..

2.5 **TITLE:** A Machine Learning Model for Garbage Classification using YOLOv4

AUTHOR: Lee, K., Kim, S., & Park, J.

YEAR: 2022

DESCRIPTION:

This study introduces GarbageNet, a machine learning model based on YOLOv4, designed for garbage classification. The research includes the creation of a large-scale dataset containing images of waste materials captured from waste bins and disposal areas. The YOLOv4 model is trained on this dataset to accurately detect and categorize garbage, contributing to smarter waste management practices..

2.6 **TITLE:** YOLO-based Trash Classification for Smart Bins in Smart Cities

AUTHOR: Chen, Y., Liu, H., & Zhang, Y.

YEAR: 2021

DESCRIPTION:

This research proposes a YOLO-based trash classification system for smart bins in smart cities. The study focuses on developing an optimized YOLO model for real-time garbage detection, enabling smart bins to segregate waste into appropriate categories automatically. The authors highlight the potential of this system in enhancing waste recycling efforts and promoting.

2.7 TITLE: Garbage Detection and Management using machine Learning and Internet of Things (IoT) Integration

AUTHOR: Das, A., Mishra, S., & Tiwari, A.

YEAR: 2020

DESCRIPTION:

This study presents a comprehensive approach to garbage detection and management by integrating machine learning, YOLO algorithm, and Internet of Things (IoT) technologies. The authors propose a scalable and interconnected system that uses YOLO for garbage detection, IoT devices for data gathering, and cloud-based analytics for waste management optimization.

CHAPTER-3

PROBLEM DEFINITION

3.1 EXISTING SYSTEM

There were no specific commercial or widely-known systems exclusively dedicated to garbage detection using the YOLO algorithm for waste management. However, research and development in this area were underway, and several academic projects and prototypes were exploring the application of YOLO and other machine learning techniques for garbage detection and waste management. Since my information might be outdated, it's possible that there have been advancements or commercial implementations in this domain..

3.1.1 DISADVANTAGE

- It is not perfect at detecting objects at different scales.
- Struggles to detect small object.

3.2 PROPOSED SYSTEM

The proposed system aims to revolutionize waste management practices by leveraging machine learning and the YOLO (You Only Look Once) algorithm for real-time garbage detection. The traditional waste management methods are often inefficient and lead to environmental degradation. This smart solution will enable automated and accurate identification of various waste materials, promoting effective waste segregation, recycling, and optimized waste collection route.

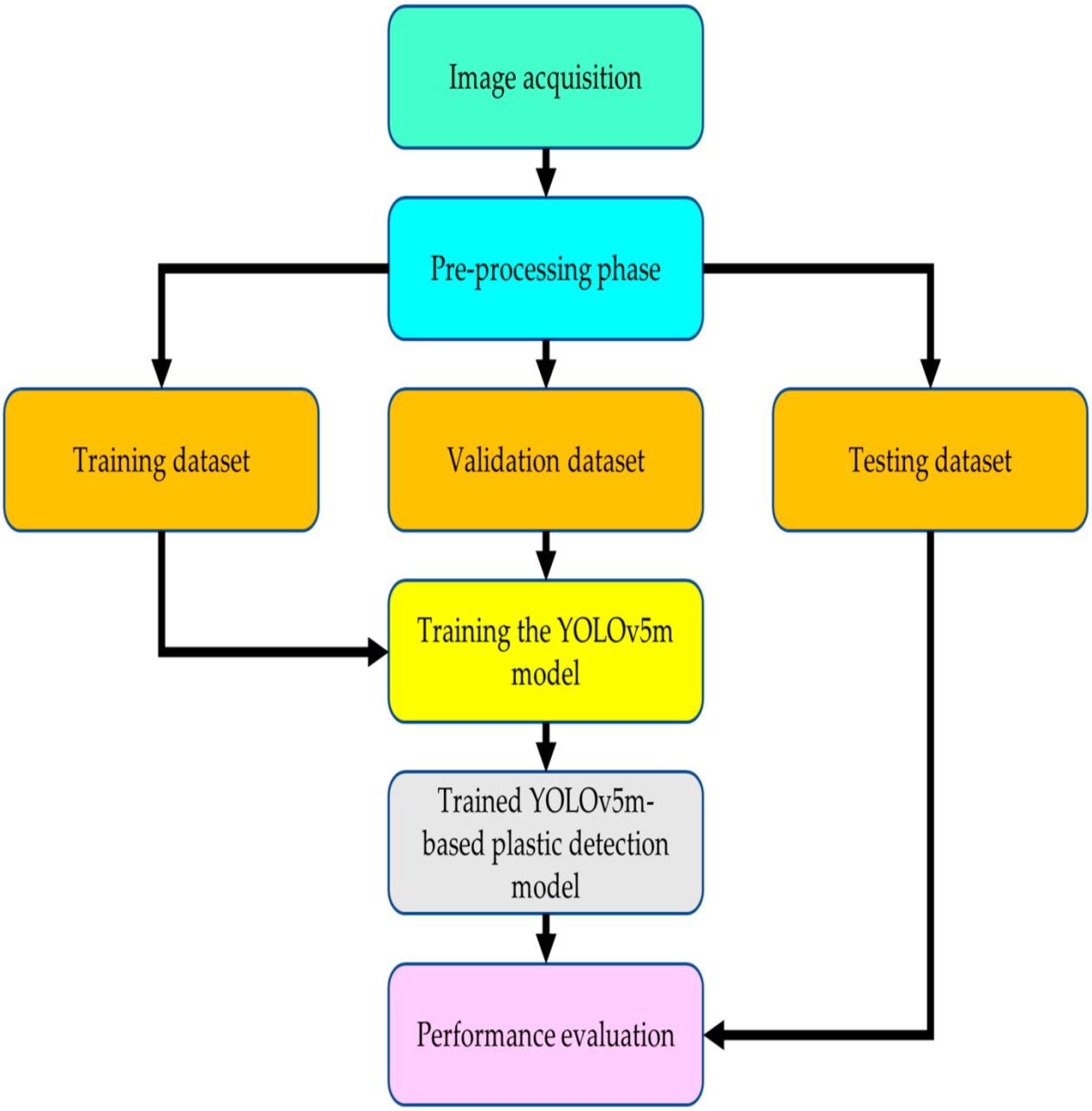


FIGURE 3.1 BLOCK DIAGRAM

3.2.1 ADVANTAGE

- Effective, accurate and adaptive
- Requires no human supervision
- Real time monitoring

3.2.2 APPLICATION

- This system mainly developed to detect and classify waste.
- It can frequently used in urban areas.

CHAPTER 4

MODULE IMPLEMENTATION

4.1 MODULE LIST

- DATASET COLLECTION
- OBJECT DETECTION
- IMAGE PREPROCESSING
- VISUALIZATION

4.2 MODULE DESCRIPTION

4.2.1 DATASET COLLECTION

Collecting a diverse and well-annotated dataset is crucial for training an accurate garbage detection model using the YOLO algorithm. The dataset should cover various garbage objects, different environmental conditions, and diverse backgrounds to ensure the model's generalization capability. Here's a step-by-step guide to collecting a garbage detection dataset for YOLO.

- Validation
- Testing
- Image

4.2.2 OBJECT DETECTION

Object detection is a computer vision task that involves identifying and localizing objects of interest within an image or a video. The goal is to create an algorithm or model capable of recognizing multiple object classes and drawing bounding boxes around them to indicate their positions.

Object detection models are typically based on deep learning algorithms, particularly Convolutional Neural Networks (CNNs). Popular object detection architectures include Faster R-CNN, YOLO (You Only Look Once), and SSD (Single Shot Multibox Detector).

These models are trained on large datasets containing annotated images, where objects of interest are labeled with bounding boxes and associated class labels. Once trained, the model can be used to detect objects in new and unseen images, making it a valuable tool for a wide range of real-world applications

4.2.3 IMAGE PREPROCESSING

Image preprocessing is a crucial step in computer vision and image processing tasks. It involves applying a series of operations or transformations to input images to improve their quality, reduce noise, and make them more suitable for analysis by machine learning algorithms or other computer vision techniques.

4.2.4 VISUALIZATION

Visualization refers to the process of representing data, information, or ideas through graphical or visual elements. The primary goal of visualization is to present complex or abstract information in a more understandable and interpretable format. It allows humans to perceive patterns, trends, and relationships in the data, making it easier to draw insights and make informed decisions.

Visualization tools and libraries, such as Tableau, matplotlib, ggplot2, D3.js, and Plotly, are widely used to create a wide range of visualizations tailored to specific needs. Choosing the appropriate visualization technique depends on the data, the message to be conveyed, and the target audience. Well-designed visualizations can significantly enhance data understanding and communication.

CHAPTER 5

SOFTWARE REQUIREMENT

H/W SYSTEM CONFIGURATION

- processor - Pentium – IV
- RAM - 4 GB (min)
- Hard Disk - 20 GB

S/W SYSTEM CONFIGURATION

- Operating System : Windows 7 or 8
- Software : python Idle

5.1 SOFTWARE ENVIRONMENT

5.1.1 PYTHON TECHNOLOGY

Python is an interpreted, high-level, general-purpose programminglanguage. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

PYTHON PROGRAMING LANGUAGE

Python is a multi-paradigm programming language. Object- oriented and structured programming are fully supported, and many of its features support functional programming and aspect- oriented programming (including by metaprogramming and met objects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python packages with a wide range of functionality, including:

- Easy to Learn and Use
- Expressive Language
- Interpreted Language
- Cross-platform Language
- Free and Open Source
- Object-Oriented Language
- Extensible
- Large Standard Library
- GUI Programming Support
- Integrated

Python uses dynamic typing and a combination of reference countingand a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has madeit particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language witha large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not usecurly brackets to delimit

blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than C or Pascal.

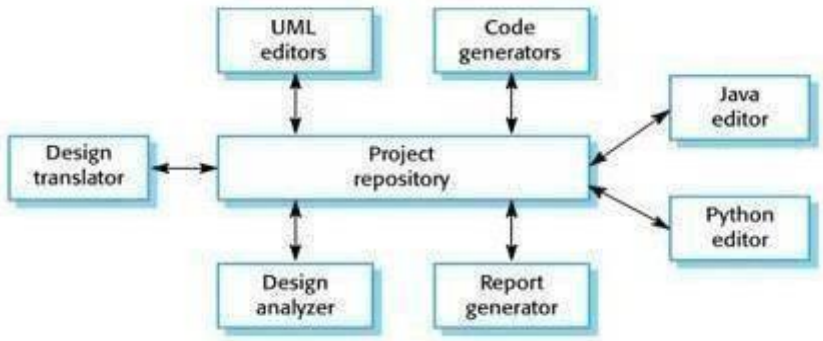


FIGURE 5.1 A REPOSITORY ARCHITECTURE FOR AN IDE

Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrastto Perl's "there is more than one way to do it" motto, Python embraces a "there should be one and preferably only one obvious way todo it" designphilosophy. Alex Martelli, a Fellow at the Python Software Foundation and Python book author, writes that "To describe something as 'clever' is not considered a compliment in the Python culture."

Python's developers strive to avoid premature optimization, and rejec patches to non-critical parts of the Python reference implementationthat would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler. Python is also available, which translates a Python script into C and makes direct C-level API calls into the Pythoninterpreter.

An important goal of Python's developers is keeping it fun to use. This is

reflected in the language's name a tribute to the British comedy group Monty Python and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a famous Monty Python sketch) instead of the standard foo and bar.

Python uses duck typing and has typed objects but untyped variable names. Type constraints are not checked at compile time;rather, operations on an object may fail, signifying that the given objectis not of a suitable type. Despite being dynamically typed, Python is strongly typed, forbidding operations that are not well-defined (for example, adding a number to a string) rather than silently attempting to make sense of them.

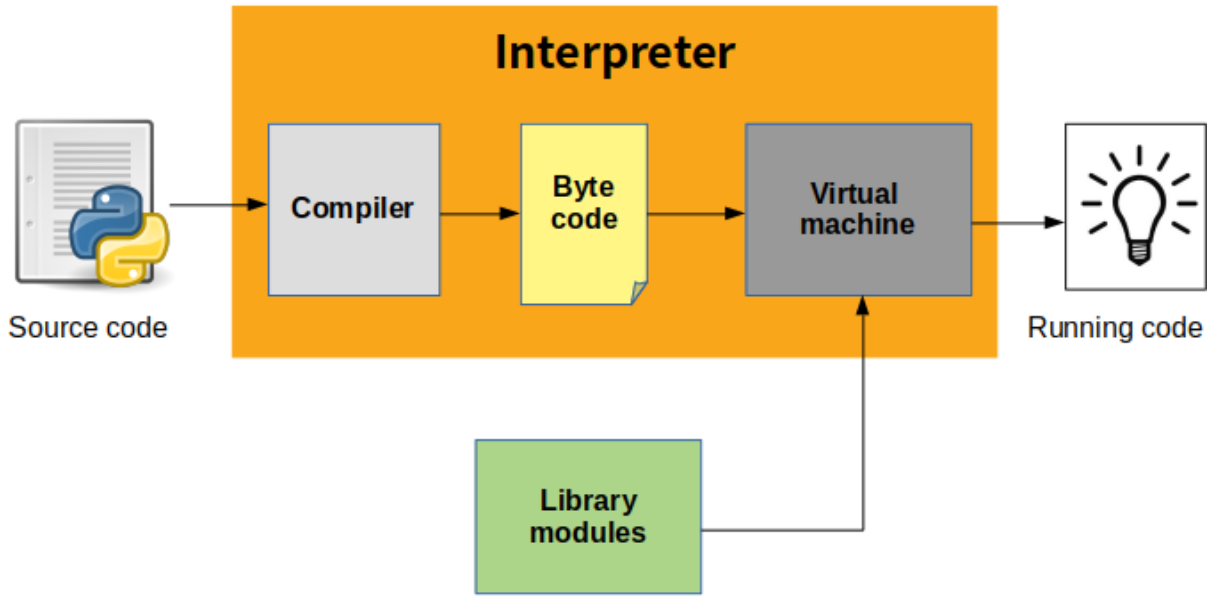


FIGURE 5.2 INTERPRETER

THE PYTHON PLATFORM

The platform module in Python is used to access the underlying platform's data, such as, hardware, operating system, and interpreter version information. The platform module includes tools to see the platform's hardware, operating system, and interpreter version information where the program is running.

There are four functions for getting information about the current Python interpreter. python_version() and python_version_tuple() return different forms of the interpreter version with major, minor, and patch level components. python_compiler() reports on the compiler used to build the interpreter. And python_build() gives a version string for the build of the interpreter.

Platform() returns string containing a general purpose platformidentifier. The function accepts two optional Boolean arguments. If aliased is true, the names in the return value are converted from a formal name to their more common form. When terse is true, returns a minimal value with some parts dropped.

What does python technology do?

Python is quite popular among programmers, but the practice shows that business owners are also Python development believers and for good reason. Software developers love it for its straightforward syntax and reputation as one of the easiest programming languages to learn. Business owners or CTOs appreciate the fact that there‟s a framework for pretty much anything – from web apps to machine learning.

Moreover, it is not just a language but more a technology platform that has come together through a gigantic collaboration from thousands of individual professional developers forming a huge and peculiar community of aficionados.

So what are the tangible benefits the language brings to those who decided to use it as a core technology? Below you will find just some of those reasons.

PRODUCTIVITY AND SPEED

It is a widespread theory within development circles that developing Python applications is approximately up to 10 times faster than developing the same application in Java or C/C++. The impressive benefit in terms of time saving can be

explained by the clean object- oriented design, enhanced process control capabilities, and strong integration and text processing capacities. Moreover, its own unit testing framework contributes substantially to its speed and productivity.

PYTHON IS POPULAR FOR WEB APPS

Web development shows no signs of slowing down, so technologies for rapid and productive web development still prevail within the market. Along with JavaScript and Ruby, Python, with its most popular web framework Django, has great support for building web apps and is rather popular within the web development community.

OPEN-SOURCE AND FRIENDLY COMMUNITY

As stated on the official website, it is developed under an OSI-approved open source license, making it freely usable and distributable. Additionally, the development is driven by the community, actively participating and organizing conference, meet-ups, hackathons, etc. fostering friendliness and knowledge-sharing.

PYTHON IS QUICK TO LEARN

It is said that the language is relatively simple so you can get pretty quick results without actually wasting too much time on constant improvements and digging into the complex engineering insights of the technology. Even though Python programmers are really in high demandthese days, its friendliness and attractiveness only help to increase number of those eager to master this programming language.

BROAD APPLICATION

It is used for the broadest spectrum of activities and applications for nearly all possible industries. It ranges from simple automation tasks to gaming, web development, and even complex enterprise systems. These are the areas where this technology is still the king with no or little competence:

- Machine learning as it has a plethora of libraries implementing machine learning algorithms.
- Web development as it provides back end for a website or an app.
- Cloud computing as Python is also known to be among one of the most popular cloud-enabled languages even used by Google in numerous enterprise-level software apps.
- Scripting.
- Desktop GUI applications.

5.1.2 AST NODES

The compiler.ast module is generated from a text file that describeseach node type and its elements. Each node type is represented as a classthat inherits from the abstract base class compiler.ast.Node and defines aset of named attributes for child nodes.

class compiler.ast.Node

The Node instances are created automatically by the parser generator. The recommended interface for specific Node instances is to use the public attributes to access child nodes. A public attribute may be bound to a single node or to a sequence of nodes, depending on the Nodetype. For example, the bases attribute of the Class node, is bound to a listof base class nodes, and the doc attribute is bound to a single node.

Each Node instance has a lineno attribute which may be None. XXX Notsure what the rules are for which nodes will have a useful lineno.

PYTHON COMPILER

The Python compiler package is a tool for analyzing Python sourcecode and generating Python bytecode. The compiler contains libraries to generate an abstract

syntax tree from Python source code and to generatePython bytecode from the tree.

The compiler package is a Python source to bytecode translator written in Python. It uses the built-in parser and standard parser module to generate a concrete syntax tree. This tree is used to generate an abstract syntax tree (AST) and then Python bytecode

The full functionality of the package duplicates the built-in compiler provided with the Python interpreter. It is intended to match its behavior almost exactly. Why implement another compiler that does the same thing? The package is useful for a variety of purposes. It can be modified more easily than the built-in compiler. The AST it generates is useful for analyzing Python source code.

THE BASIC INTERFACE

The top-level of the package defines four functions. If you import compiler, you will get these functions and a collection of modules contained in the package.

compiler.parse(buf)

Returns an abstract syntax tree for the Python source code in buf. The function raises Syntax Error if there is an error in the source code. The return value is a compiler.ast. Module instance that contains the tree.

compiler.parseFile(path)

Return an abstract syntax tree for the Python source code in the file specified by path. It is equivalent to parse(open(path).read()).

LIMITATIONS

There are some problems with the error checking of the compiler package. The interpreter detects syntax errors in two distinct phases. One set of errors is detected by the interpreter‟s parser, the other set by the compiler. The compiler package relies on the interpreter‟s parser, so it get the first phases of error checking for free. It implements the secondphase itself, and that implementation is incomplete.

For example, the compiler package does not raise an error if a name appears more than once in an argument list: def f(x, x): ...

A future version of the compiler should fix these problems.

PYTHON ABSTRACT SYNTAX

The compiler.ast module defines an abstract syntax for Python. In the abstract syntax tree, each node represents a syntactic construct. The root of the tree is Module object.

The abstract syntax offers a higher level interface to parsed Python source code. The parser module and the compiler written in C for the Python interpreter use a concrete syntax tree. The concrete syntax is tied closely to the grammar description used for the Python parser. Instead ofa single node for a construct, there are often several levels of nested nodes that are introduced by Python‟s precedence rules.

The abstract syntax tree is created by the compiler. transformer module. The transformer relies on the built-in Python parser to generatea concrete syntax tree. It generates an abstract syntax tree from the concrete tree.

The transformer module was created by Greg Stein and Bill Tutt for an experimental Python-to-C compiler. The current version contains a number of modifications and improvements, but the basic form of the abstract syntax and of the transformer are due to Stein and Tutt.

ALL NODE OBJECTS OFFER THE FOLLOWING METHODS

getChildren()

Returns a flattened list of the child nodes and objects in the order they occur. Specifically, the order of the nodes is the order in which they appear in the Python grammar. Not all of the children are Node instances. The names of functions and classes, for example, are plain strings.

getChildNodes()

Returns a flattened list of the child nodes in the order they occur. This method is like getChildren(), except that it only returns those children that are Node instances.

The While node has three attributes: test, body, and else_. (If the natural name for an attribute is also a Python reserved word, it can‟t be used as an attribute name. An underscore is appended to the word to make it a legal identifier, hence else_ instead of else.)

The if statement is more complicated because it can include several tests.
The If node only defines two attributes: tests and else_. The tests attribute is a sequence of test expression, consequent body pairs. There is one pair for each if/elif clause. The first element of the pair is the test expression. The second elements is a Stmt node that contains the code toexecute if the test is true.

The getChildren() method of If returns a flat list of child nodes. If there are three if/elif clauses and no else clause, then getChildren() will return a list of six elements: the first test expression, the first Stmt, the second text expression, etc.

The following table lists each of the Node subclasses defined in compiler.ast and each of the public attributes available on their instances. The values of most of the attributes are themselves Node instances or sequences of instances. When the value is something other than an instance, the type is noted in the comment. The attributes are listed in the order in which they are returned by getChildren() and getChildNodes().

5.2 DEVELOPMENT ENVIRONMENTS

Most Python implementations (including CPython) include a read–eval–print loop (REPL), permitting them to function as a command line interpreter for which the user enters statements sequentially and receives results immediately.Other shells, including IDLE and IPython, add further abilities such as auto-completion,

session state retention and syntax highlighting.

5.3 IMPLEMENTATIONS

REFERENCE IMPLEMENTATION

CPython is the reference implementation of Python. It is written in C, meeting the C89 standard with several select C99 features. It compiles Python programs into an intermediate bytecode which is then executed by its virtual machine. CPython is distributed with a large standard library written in a mixture of C and native Python. It is available for many platforms, including Windows and most modern Unix-like systems. Platform portability was one of its earliest priorities.

OTHER IMPLEMENTATIONS

PyPy is a fast, compliant interpreter of Python 2.7 and 3.5. Its just-in- time compiler brings a significant speed improvement over CPython but several libraries written in C cannot be used with it.

Stackless Python is a significant fork of CPython that implements microthreads; it does not use the C memory stack, thus allowing massively concurrent programs. PyPy also has a stackless version.

MicroPython and CircuitPython are Python 3 variants optimized for microcontrollers. This includes Lego Mindstorms EV3.

RustPython is a Python 3 interpreter written in Rust.

UNSUPPORTED IMPLEMENTATIONS

Other just-in-time Python compilers have been developed, but are now unsupported: Google began a project named Unladen Swallow in 2009, with the aim of speeding up the Python interpreter five-fold by using the LLVM, and of improving its multithreading ability to scale to thousands of cores, while ordinary implementations

suffer from the global interpreter lock.

Psyco is a just-in-time specialising compiler that integrates with CPython and transforms bytecode to machine code at runtime. The emitted code is specialized for certain data types and is faster than standard Python code.

In 2005, Nokia released a Python interpreter for the Series 60 mobile phones named PyS60. It includes many of the modules from the CPython implementations and some additional modules to integrate withthe Symbian operating system. The project has been kept up-to-date to run on all variants of the S60 platform, and several third-party modules are available. The Nokia N900 also supports Python with GTK widget libraries, enabling programs to be written and run on the target device.

CROSS-COMPILERS TO OTHER LANGUAGES

There are several compilers to high-level object languages, with either unrestricted Python, a restricted subset of Python, or a language similar to Python as the source language:

- Jython enables the use of the Java class library from a Python program.
- IronPython follows a similar approach in order to run Python programs on the .NET Common Language Runtime.
- The RPython language can be compiled to C, and is used to buildthe PyPy interpreter of Python.
- Pyjs compiles Python to JavaScript.

- Cython compiles Python to C and C++.
- Numba uses LLVM to compile Python to machine code.
- Pythran compiles Python to C++.
- Somewhat dated Pyrex (latest release in 2010) and Shed Skin(latest release in 2013) compile to C and C++ respectively.
- Google's Grumpy compiles Python to Go.
- MyHDL compiles Python to VHDL.
- Nuitka compiles Python into C++.

PERFORMANCE

A performance comparison of various Python implementations on a non-numerical (combinatorial) workload was presented at EuroSciPy '13.

API DOCUMENTATION GENERATORS

Python API documentation generators include:

- Sphinx
- Epydoc
- HeaderDoc
- Pydoc

USES

Python has been successfully embedded in many software products as ascripting language, including in finite element method software such as Abaqus, 3D parametric

modeler like FreeCAD, 3D animation packages such as 3ds Max, Blender, Cinema 4D, Lightwave, Houdini, Maya,modo, MotionBuilder, Softimage, the visual effects compositor Nuke, 2D imaging programs like GIMP, Inkscape, Scribus and Paint Shop Pro, and musical notation programs like scorewriter and capella. GNU Debugger uses Python as a pretty printer to show complex structures such as C++ containers. Esri promotes Python as the best choice for writing scripts in ArcGIS. It has also been used in several video games, and has been adopted as first of the three available programming languages in Google App Engine, the other two being Java and Go.

Python is commonly used in artificial intelligence projects with thehelp of libraries like TensorFlow, Keras and Scikit-learn. As a scripting language with modular architecture, simple syntax and rich text processing tools, Python is often used for natural language processing.

Many operating systems include Python as a standard component. It ships with most Linux distributions, AmigaOS 4, FreeBSD (as a package), NetBSD, OpenBSD (as a package) and macOS and can be used from the command line (terminal). Many Linux distributions use installers written in Python: Ubuntu uses the Ubiquity installer, while Red Hat Linux and Fedora use the Anaconda installer. Gentoo Linux uses Python in its package management system, Portage.

Python is used extensively in the information security industry, including in exploit development.

Most of the Sugar software for the One Laptop per Child XO, now developed at Sugar Labs, is written in Python. The Raspberry Pi single- board computer project has adopted Python as its main user- programming language.

LibreOffice includes Python, and intends to replace Java with Python. Its Python Scripting Provider is a core feature since Version 4.0 from 7 February 2013.

PANDAS

In computer programming, pandas is a software library written for the Python programming language for data manipulation and analysis. Inparticular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.

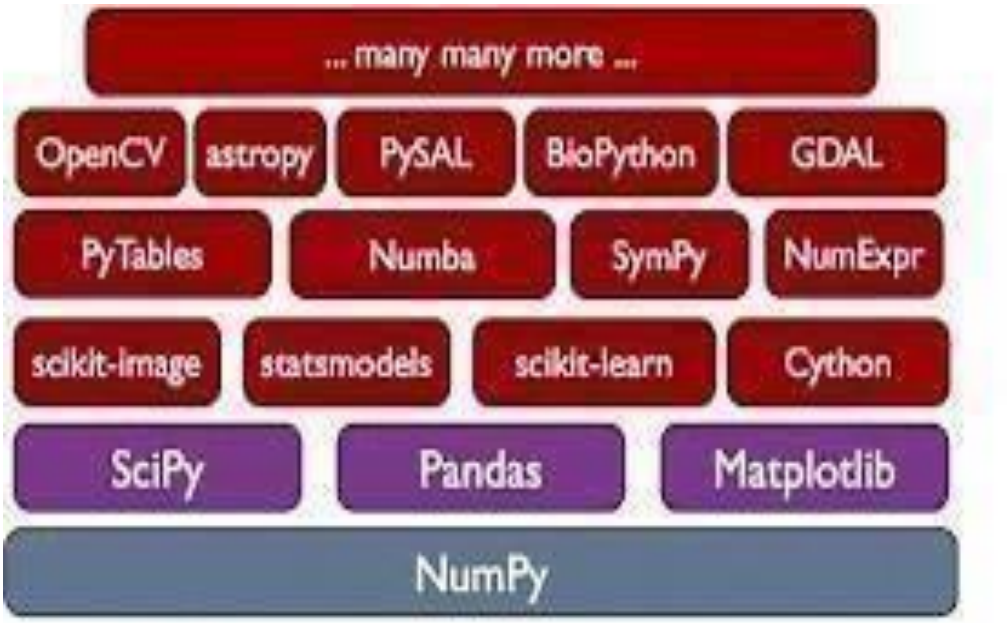


FIGURE 5.3 PANDAS

LIBRARY FEATURES

- Data Frame object for data manipulation with integrated indexing.
- Tools for reading and writing data between in-memory data structures and different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of data sets.

- Label-based slicing, fancy indexing, and sub setting of large datasets.
- Data structure column insertion and deletion.
- Group by engine allowing split-apply-combine operations on datasets.
- Data set merging and joining.
- Hierarchical axis indexing to work with high-dimensional data in alower-dimensional data structure.
- Time series-functionality: Date range generation and frequencyconversion, moving window statistics, moving window linear regressions, date shifting and lagging.
- Provides data filtration.

5.4 CSV READER

CSV (Comma Separated Values) is a simple file format used to store tabular data, such as a spreadsheet or database. A CSV file stores tabular data (numbers and text) in plain text. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format. For working CSV files in python, there is an inbuilt module called csv.

CHAPTER 6

SYSTEM DESIGN

6.1 UML DIAGRAMS

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

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

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

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

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

GOALS

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

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

6.2 USE CASE DIAGRAM

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

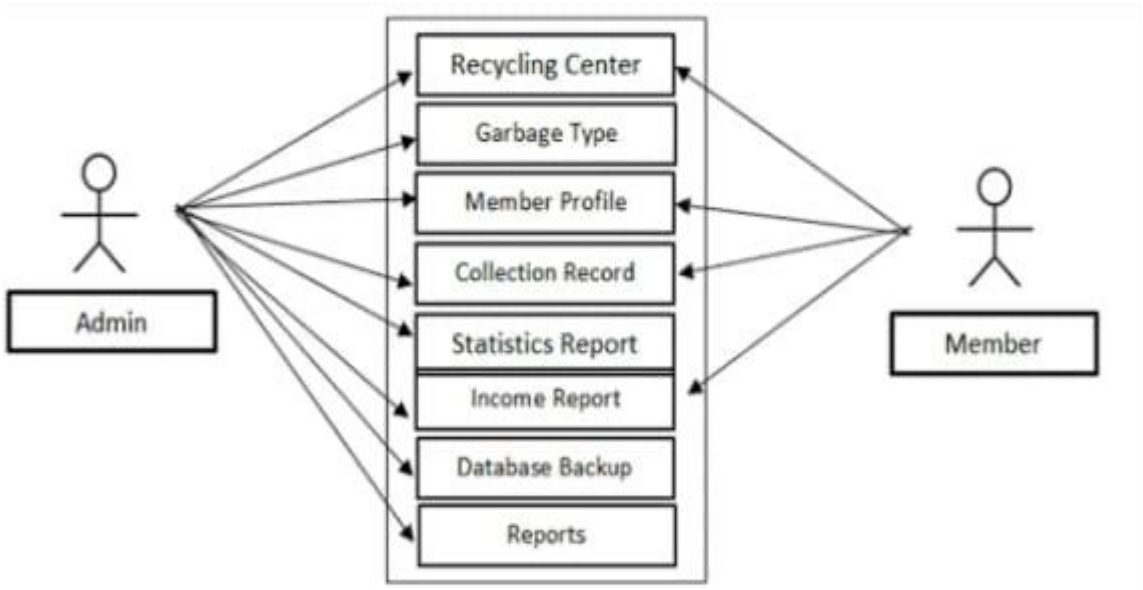


FIGURE 6.1 USE CASE DIAGRAM

6.3 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. Itexplains which class contains information.

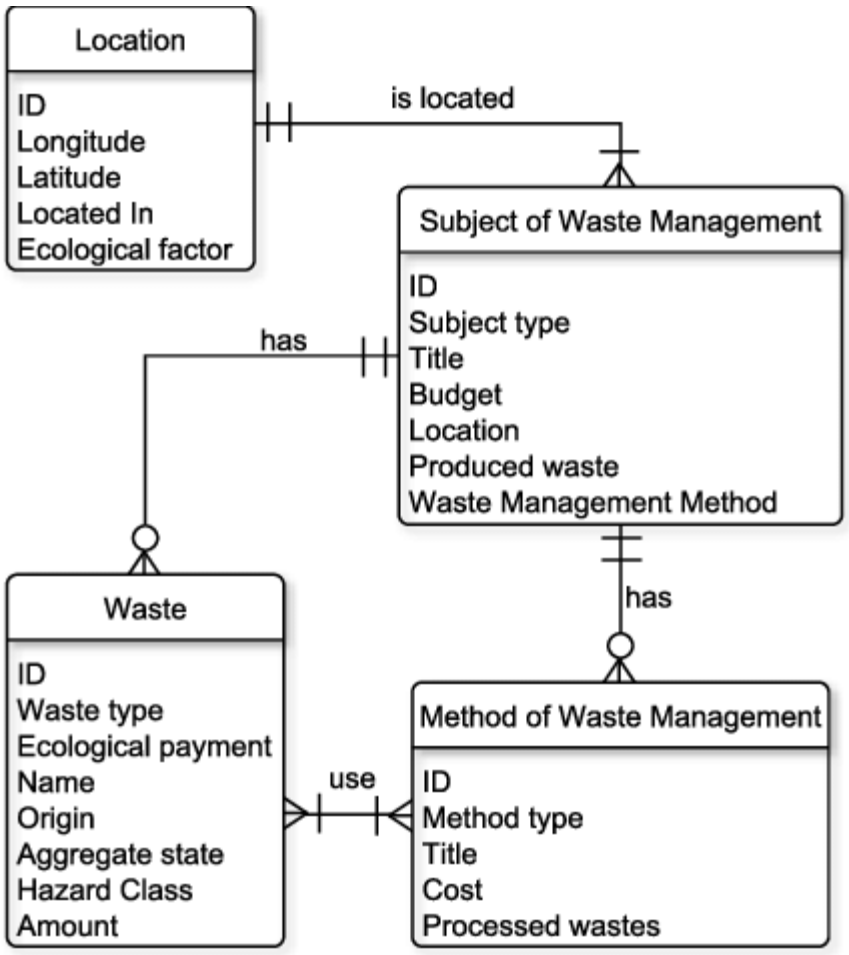


FIGURE 6.2 CLASS DIAGRAM

6.3 SEQUENCE

Sequence diagrams, commonly used by developers, model the interactions between objects in a single use case. They illustrate how thedifferent parts of a system interact with each other to carry out a function, and the order in which the interactions occur when a particularuse case is executed.

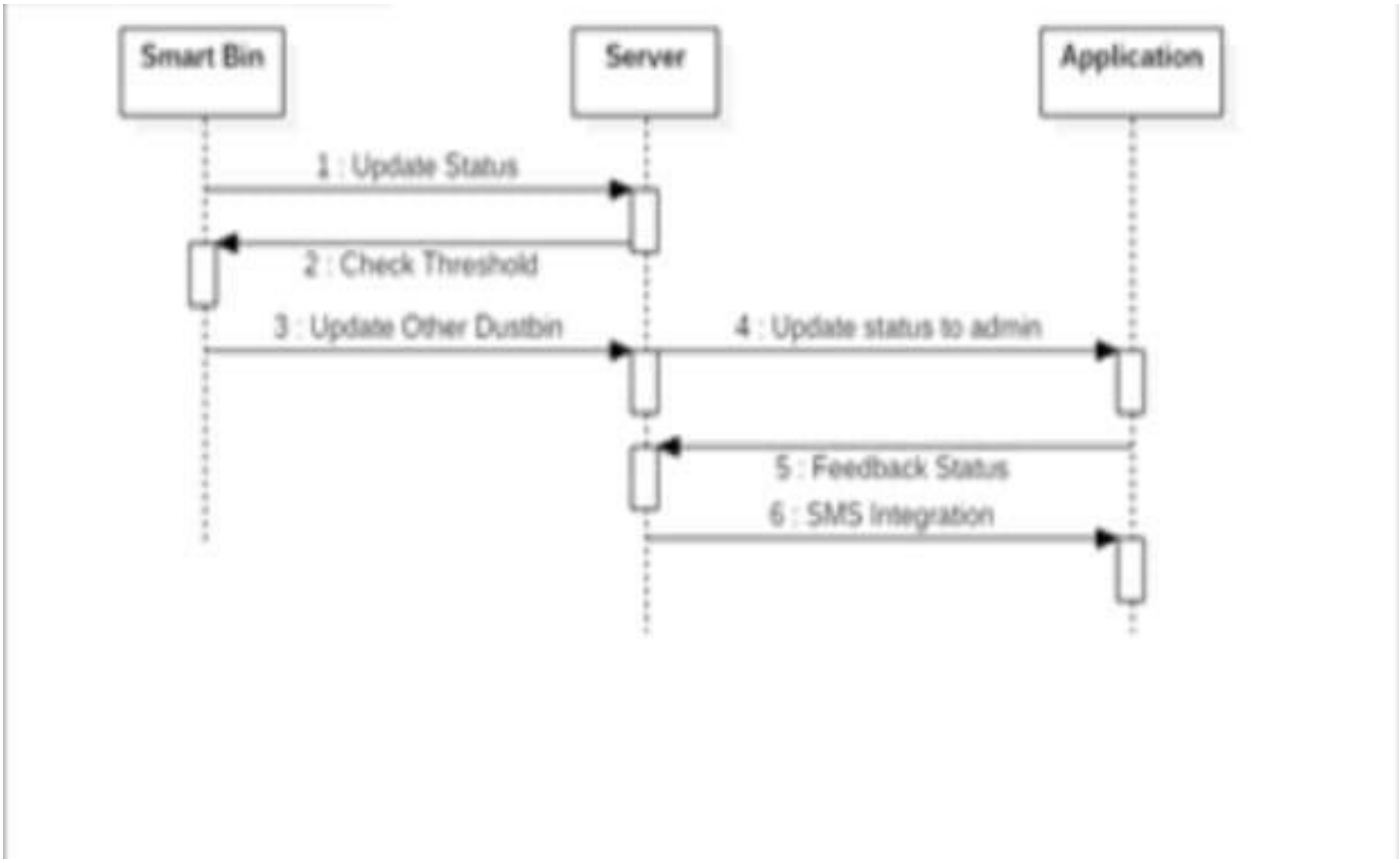


FIGURE 6.3 SEQUENCE DIAGRAM

6.4 DEPLOYMENT

Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware. UML is mainly designed to focus on the software artifacts of a system. However, these two diagrams are special diagrams used to focus on software and hardware components.

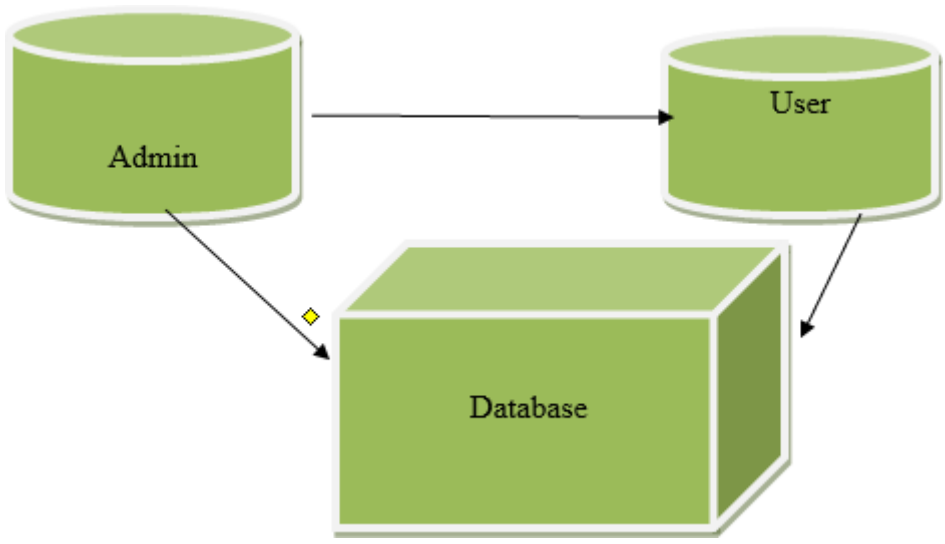


FIGURE 6.4 DEPLOYMENT DIAGRAM

6.5 DATA FLOW DIAGRAM

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input datato the system, various processing carried out on this data, and the output data is generated by this system.

2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphicaltechnique that depicts information flow and the transformations that are applied as data moves from input to output.

4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functionaldetail.

CHAPTER 7

SYSTEM TESTING

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, sub-assemblies, 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.

7.1 TYPES OF TESTS

7.1.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flowshould 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.

7.1.2 INTEGRATION TESTING

Integration tests are designed to test integrated softwarecomponents to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens orfields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.1.3 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.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematiccoverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.2 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 processdescriptions and flows, emphasizing pre-driven process links and integration points.

7.2.1 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

7.2.2 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested.

Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

7.3 UNIT TESTING

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

TEST STRATEGY AND APPROACH

Field testing will be performed manually and functional tests will be written in detail.

TEST OBJECTIVES

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

FEATURES TO BE TESTED

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

7.4 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.

TEST RESULTS: All the test cases mentioned above passed successfully. No defects encountered.

7.5 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.

TEST RESULTS: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER-8
CONCLUSION

8.1 CONCLUSION

Garbage detection in machine learning using the YOLO algorithm offers a transformative solution to the pressing challenges of waste management. Through the integration of cutting-edge computer vision techniques and real-time object detection capabilities, this approach has the potential to revolutionize waste collection, recycling, and environmental sustainability.

By creating a comprehensive dataset and training the YOLO model on various waste categories, the system can accurately and efficiently identify different types of garbage in real-world scenarios. The benefits of such a system multifaceted

1. Improved Waste Management: The ability to identify and classify garbage items in real-time enables waste management authorities to optimize collection routes and schedules. This leads to more efficient waste disposal practices, reducing operational costs and resource wastage.

2. Enhanced Recycling Efforts: Proper waste segregation facilitated by the YOLO-based garbage detection system improves the quality of recyclable materials, promoting recycling initiatives and minimizing the burden on landfills.

APPENDIX A (SAMPLE CODING)

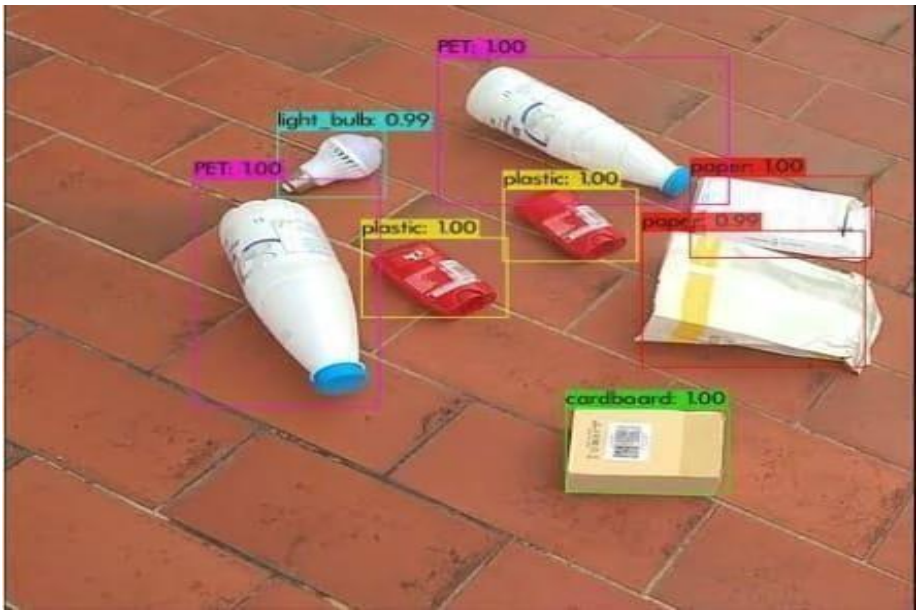
```
import cv2
import numpy as np

# Load the YOLO model and configuration files
net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")

# Load the class labels
classes = []
with open("waste_classes.txt", "r") as f:
    classes = [line.strip() for line in f.readlines()]

# Set the minimum confidence level and threshold for non-maximum suppression
confidence_threshold = 0.5
nms_threshold = 0.4

# Load the image and prepare it for object detection
image = cv2.imread("test_image.jpg")
height, width, _ = image.shape

# Create a blob from the input image
blob = cv2.dnn.blobFromImage(image, 1/255.0, (416, 416), swapRB=True, crop=False)

# Set the input blob for the network
```

```
net.setInput(blob)

# Run forward pass through the network
output_layers = net.getUnconnectedOutLayersNames()
layer_outputs = net.forward(output_layers)

# Initialize lists to store detected class IDs, confidences, and bounding box coordinates
class_ids = []
confidences = []
boxes = []

# Iterate over each output layer
for output in layer_outputs:
    # Iterate over each detection
    for detection in output:
        scores = detection[5:]
        class_id = np.argmax(scores)
        confidence = scores[class_id]

        # Filter out weak predictions
        if confidence > confidence_threshold:
            # Scale the bounding box coordinates to the original image size
            center_x = int(detection[0] * width)
            center_y = int(detection[1] * height)
```

```
            w = int(detection[2] * width)
            h = int(detection[3] * height)
            x = int(center_x - w / 2)
            y = int(center_y - h / 2)

            class_ids.append(class_id)
            confidences.append(float(confidence))
            boxes.append([x, y, w, h])

# Perform non-maximum suppression to remove redundant overlapping boxes
indices = cv2.dnn.NMSBoxes(boxes, confidences, confidence_threshold, nms_threshold)

# Iterate over the selected bounding boxes and draw them on the image
for i in indices:
    i = i[0]
    x, y, w, h = boxes[i]
    class_id = class_ids[i]
    label = f"{classes[class_id]}: {confidences[i]:.2f}"
    color = (0, 255, 0) # Green color for the bounding box
    cv2.rectangle(image, (x, y), (x + w, y + h), color, 2)
    cv2.putText(image, label, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)

# Display the resulting image
cv2.imshow("Garbage Waste Detection", image)
```

```
cv2.waitKey(0)
cv2.destroyAllWindows()
```

APPENDIX B (SCREEN SHOTS)



REFERENCES

[1] H. Xiao, Z. Li, X. Jia et al, Chapter 2 - Waste to energy in a circular economy approach for better sustainability: a comprehensive review and SWOT analysis [J], Waste-to-Energy, 2020, 23-43.

[2] S. B. Atitallah, M. Driss, W Boulila et al, Compliance with household solid waste management in rural villages in developing countries [J], Computer Science Review, 2020, 38:100303.

[3] F. Wang, Z. Cheng, A. Reisner et al. Compliance with household solid waste management in rural villages in developing countries [J], Journal of Cleaner Production, 2018, 202: 293-298.

[4] D. Gabriel, J. Toutouh, S. Nesmachnow. Exact and heuristic approaches for multi-objective garbage accumulation points loca-tion in real scenarios [J], 2020, 105:467-481.

[5] N. Leeabai, C. Areeprasert, C. Khaobang. The effects of color preference and noticeability of trash bins on waste collection performance and waste-sorting behaviors [J], Waste Management, 2021, 121:153-163.