A

Major Project Report

on

HERITAGE IDENTIFICATION OF MONUMENTS USING DEEP LEARNING TECHNIQUES

Submitted in Partial Fulfillment of
the Requirements for the Degree
of

Bachelor of Engineering

in

Computer Engineering

to

**Kavayitri Bahinabai Chaudhari
North Maharashtra University, Jalgaon**

Submitted by

**Lina Shamkant Baviskar
Veena Anil Patil
Harshada Sanjiv Bundelkar**

Under the Guidance of

Mr. S. H. Rajput

DEPARTMENT OF COMPUTER ENGINEERING
SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY,
BAMBHORI, JALGAON - 425 001 (MS)
2023 - 2024

SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY, BAMBHORI, JALGAON - 425 001 (MS)

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the major project entitled *Heritage Identification of Monuments using Deep learning Techniques*, submitted by

Lina Shamkant Baviskar
Veena Anil Patil
Harshada Sanjiv Bundelkar

in partial fulfillment of the degree of *Bachelor of Engineering* in *Computer Engineering* has been satisfactorily carried out under my guidance as per the requirement of Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon.

Date: May 2, 2024
Place: Jalgaon

Mr. S. H. Rajput
Guide

Dr. Manoj E. Patil
Head

Prof. Dr. G.K.Patnaik
Principal

Acknowledgement

We would like to express our deep gratitude and sincere thanks to all who helped us in completing this project report successfully. Many thanks to almighty God who gave us the strength to do this. Our sincere thanks to Principal Prof. Dr. Girish. K. Patnaik for providing the facilities to complete this Project report. We would like to express our gratitude and appreciation to all those who gave us the possibility to complete this report. A special thanks to Prof. Dr. Manoj E. Patil, Head of the Department, whose help, stimulating suggestions and encouragement, helped us in writing this report. We would also like to thank Mr. S. H. Rajput, Project Guide, who has given his full effort in guiding us and achieving the goal as well as his encouragement to maintain the progress in track. I am also sincerely thankful to prof. Aakash D. Waghmare, Incharge of Project, for her valuable suggestions and guidance. We would also like to appreciate the guidance given by other supervisor that has improved our presentation skills by their comments and tips. Last but not the least, we are extremely thankful to our parents and friends without whom it could not reach its successful completion.

Lina Shamkant Baviskar
Veena Anil Patil
Harshda Sanjiv Bundelkar

Contents

List of Figures

List of Tables

Abstract

India is a nation with a plethora of cultural landmarks, including notable architectural masterpieces, 37 of which are UNESCO World Heritage master pieces. We must protect cultural heritages because they bind successive generations together over time. Architectural Designers, researchers, and travellers, etc. visit numerous historical locations, where it is frequently challenging for them to recognise and learn more about the historical significance of the monument in which they are interested. Due to the size and dependability of the information, the work of archiving, recording, and sharing the knowledge of these cultural assets is difficult. Modern machine learning and deep learning algorithms, as well as high processing computational resources, offer practical answers. The classification of Monument satellite images or photographs can be atomized by utilising the Convoluted Neural Network techniques. During our project implementation, monuments images dataset was created with the help of google earth images.

We have applied various pre-processing techniques and from pre-processed images we extracted features using feature extraction techniques such as Local Binary Patterns(LBP), Mean Standard Deviation(MSD). A model was developed using deep learning algorithms such as the Convoluted neural network(CNN). The results of our project are discussed in this paper. Initially we upload the monument satellite image, then the system recognizes the monument first then predicts whether the monument is heritage or not and gives the accuracy value as well as displays the monument image along with information of the monument.

Initially we upload the monument satellite image, then the system recognizes the monument first then predicts whether the monument is heritage or not and gives the accuracy value as well as displays the monument image along with information of the monument

Chapter 1

Introduction

Heritage of a monument has prominence as the people belonging to the various cultures, castes, creeds and religions take pride in their culturally rich heritage bestowed upon them in the form of monuments. There is a need to digitally recognize and archive the monuments as an important historical and cultural heritage site. This poses an impactful feature not only for tourists but also for the local people in order to gain knowledge regarding their own cultural heritage.

Various types of Deep Learning architectures have been used to recognize monuments and achieve a good performance. Our method leverages the rich visual features extracted from satellite images to automatically detect and classify monuments. We highlight the importance of using CNNs for heritage identification, as they can learn complex patterns and representations from large amounts of data, making them well-suited for handling the diverse and intricate visual characteristics of heritage sites. We also discuss the potential benefits and applications of our proposed approach, including assisting archaeologists, cultural heritage experts, and policy makers in heritage preservation and management. Eventually, our project strives to state the importance of preserving and maintaining information for a cultural heritage digitally.

1.1 Background

Heritage identification is like solving a historical puzzle, and monuments are the pieces that reveal the story of our past. To properly identify and appreciate these cultural treasures, we need to delve into their backgrounds. This involves unraveling the threads of history, architecture, and cultural significance. Start by examining the architectural style—whether it's Gothic, Baroque, or something else. This can be a clue to the era in which the monument was built. Look at the materials used; they can tell tales of local resources and craftsmanship. Dig into historical records to understand the purpose behind the construction. Was it a religious site, a governmental building, or a symbol of societal power? This context helps in unraveling the layers of meaning embedded in the structure. Explore the cultural motifs and symbols intricately carved into the monument. These details often reflect the values, beliefs, and narratives of the society that created them. Consider the geographical location. Monuments are often deeply connected to the landscape, whether strategically placed on a hill for defense or nestled in a vibrant city center. Lastly, don't forget the stories passed down through generations. Oral histories can provide valuable insights into the significance of a monument in the eyes of the community. In essence, heritage identification is a blend of detective work, historical analysis, and cultural appreciation—a journey through time that brings these monuments to life.

1.2 Motivation

Unveiling the hidden stories of our cultural heritage through deep learning is not just a scientific endeavor; it's a journey that breathes life into the stones and structures of our past. The motivation behind this report is fueled by the desire to transcend traditional methods of heritage identification and usher in a new era of understanding. Deep learning, with its ability to sift through vast amounts of data and discern intricate patterns, becomes the modern-day storyteller of our historical treasures. It empowers us to decode the language of architecture, unravel the secrets embedded in ancient designs, and piece together the narratives etched in stone. By employing deep learning techniques, we aim to enhance the precision and efficiency of heritage identification. This isn't just about classifying monuments; it's about capturing the essence of cultural evolution. The motivation lies in preserving our collective memory, bridging the gap between the past and the present.

1.3 Problem Definition

Developing deep learning models that can navigate the intricate web of architectural diversity, historical contexts, data variability, and interpretability challenges to accurately identify and classify monuments, contributing to the preservation and understanding of our cultural heritage.

1.4 Scope

This project scope outlines the key components and objectives for developing a deep learning-based system for the identification of monuments, aiming to contribute to the preservation and understanding of cultural heritage worldwide:

1.5 Objectives

- Develop a deep learning model capable of accurately identifying and classifying diverse monuments based on architectural features, historical contexts, and cultural significance.
- Enhance the precision and efficiency of heritage identification, transcending the limitations of traditional methods.

1.6 Selection of Life cycle model for development

The software development life cycle model selected for this project is the Agile Model. Agile approach was the first SDLC (Software Development Life Cycle) Model to be widely used in software engineering to ensure success of the project. It was developed by Ken Schwaber and Jeff Sutherland in 2001. In "The Agile" approach, the whole process of software development is divided into separate phases, typically the outcome of one phase acts as the input for the next phase sequentially. All the phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. Re-quirements for this project are well documented and fixed.

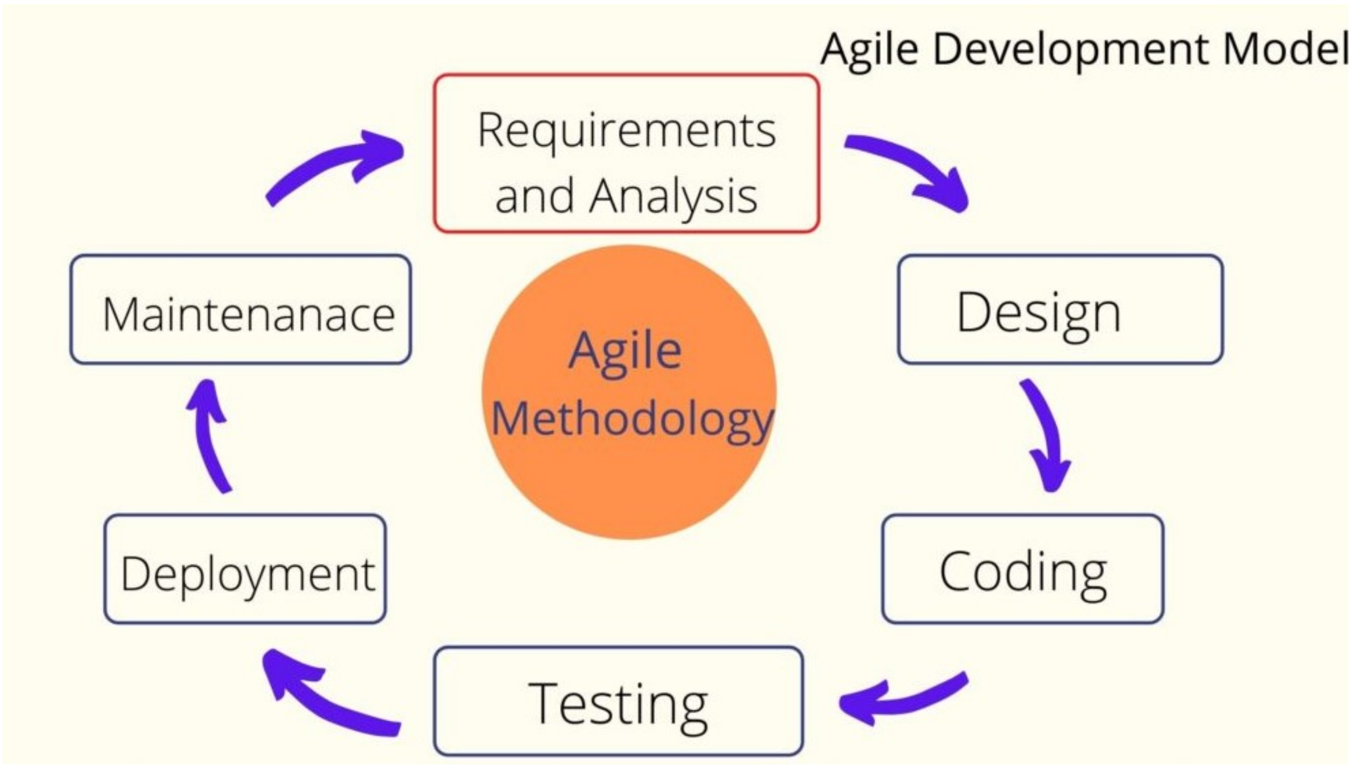


Figure 1.1: Agile Model

Agile Model is best suited model for this project.

- Because requirements are easily understandable and defined
- We can define requirements in early stage of development
- User involvement in all phases is not necessary
- Limited user's participation

1.7 Why we are using Agile Model

Agile is great for situations where requirements might evolve as you go along, which is pretty common when dealing with cutting-edge technologies like deep learning. It allows you to adapt quickly to changes, get regular feedback, and ensure that the final product meets the actual needs of the end-users. Plus, it encourages collaboration and communication, which is crucial when you're dealing with a complex project involving heritage identification and deep learning.

So, in a nutshell, Agile is like the flexible friend you want by your side when you're navigating the ever-changing landscape of heritage identification using deep learning techniques.

1.8 Organization of Report

Chapter 1: entitled as Introduction describes the details about Background, Problem Definition, Scope and Objective of the project, Identification of Software Development Process Model and Organization of report.

Chapter 2: entitled as Project Planning and Management consists of details about the Feasibility Study, Risk Analysis, Project Scheduling, Effort Allocation and Cost Estimation of the project.

Chapter 3: entitled as Analysis describes in detail, the Requirement Collection and Identification, H/w and S/w Requirements, Functional and Non-Functional Requirements and a Software Requirements Specification(SRS).

Chapter 4: includes design about System Architecture, Data Flow Diagram and various UML Diagrams.

Chapter 5: includes Coding/Implementation about Algorithm, Software and Hardware Requirement and varoious Modules in Project.

Chapter 6: includes Testing about Black Box and White Box Testing, Manual and Automated Testing and Test case Identification and Execution.

Chapter 7: includes Results and Discussion about Results and Various Future Work

1.9 Summary

As mentioned in above sections, this project aims at building an System for Monuments Recognition. The scopes, objective, etc. are as mentioned above. In the next chapter, project planning and management will be discussed.

Chapter 2

Project Planning And Management

Develop a strategy for involving local communities in the project. This could include educational programs, workshops, or community outreach to raise awareness and gather valuable insights. Establish protocols for managing and analyzing the collected data. This includes organizing archival information, digitizing records, and ensuring data integrity. Plan for the creation of a comprehensive project report that includes the methodology, findings, and recommendations. Ensure that the report is accessible to both experts and the general public. Implement a system for ongoing monitoring and evaluation of the project's progress. This will allow for adjustments to be made if needed and ensure the project's success.

The organization of this chapter is as below. Section 2.1 shows the Feasibility Study of the project. Risk Analysis of the project is represented in Section 2.2 and Project Scheduling is described in Section 2.3. Section 2.4 and 2.5 describe the Effort Allocation and Cost Estimation respectively. The Summary is mentioned in Section 2.6.

2.1 Feasibility Study

A feasibility study is an assessment of the practicality of a proposed project or system. A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the natural environment, the resources required to carry through, and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained. A well-designed feasibility study should provide a historical background of the business or project, a description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, feasibility studies precede technical development and project implementation.

A feasibility study evaluates the project's potential for success; therefore, perceived objectivity is an important factor in the credibility of the study for potential investors and lending institutions. It must therefore be conducted with an objective, unbiased approach to provide information upon which decisions can be based. Taking into consideration the technical, operational and economic feasibility as below, the project can be anticipated as feasible overall. There are few types of feasibility that exists. So, developers should take care of these feasibility and take them into consideration:

2.1.1 Technical Feasibility

This assessment is based on an outline design of system requirements, to determine whether the company has the technical expertise to handle completion of the project. At this level, the concern is whether the proposal is both technically and legally feasible (assuming moderate cost). It is an evaluation of the hardware and software and how it meets the need of the proposed system.

This project is built upon Jupyter Notebook, a simple Web-Application or CLI with Python as the programming language and can be easily hosted on cloud server. Also all the other technologies used are capable of building such a project and serve as well as maintain it for longer period of time. All the required hardware and software are easily available in the market. Hence the project is technically feasible.

2.1.2 Operational Feasibility

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

The operational feasibility assessment focuses on the degree to which the proposed development project fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes. The application is operationally feasible since it is build with the idea for integration with various existing applications and systems.

2.1.3 Economical Feasibility

Describes how much time is available to build the new system, when it can be built, whether it interferes with normal business operations, type and amount of resources required, dependencies, and developmental procedures with company revenue prospectus.

As the necessary hardware and the software are easily available in the market at low cost, the initial investment is the only cost incurred and does not need further enhancement. Hence it is economically feasible.

2.2 Risk Analysis

Risk Analysis and Management is a key project management practice to ensure that the least number of surprises occur while your project is underway. While we can never predict the future with certainty, we can apply a simple and streamlined risk management process to predict the uncertainties in the projects and minimize the occurrence or impact of these uncertainties.

This project has a very slight window for experiencing failures not in technical aspects but those functionalities involving real life interaction might be affected.

2.3 Project Scheduling

Generally, project scheduling can be stated as the estimated time required for any project from its time of beginning to the end of the project. In detail, for every task, there is a deadline because all the tasks for the completion of project are planned earlier. So that, each task is scheduled to certain time limit. In short, in project management, listing of projects milestones, activities and all from starting to end date, are considered in the project scheduling. A schedule is generally used in the project planning and management of the project with some kind of attributes as budget, task allocation and duration, resource allocation and all.

Task	Start Date	End Date
Selection of title	24 Jul 2023	11 Aug 2023
Gathering information	14 Aug 2023	22 Sep 2023
Project discussion	23 Sep 2023	11 Oct 2023
Planning/ requirement gathering and analysis	14 oct 2023	5 Nov 2023
Design	18 Nov 2023	1 Jan 2024
Coding or Implementation	1 Jan 2024	11 Mar 2024
Testing	11 Mar 2024	5 Apr 2024
Result and Discussion	7 Apr 2024	23 Apr 2024

Table 2.1: Task Scheduling for the project

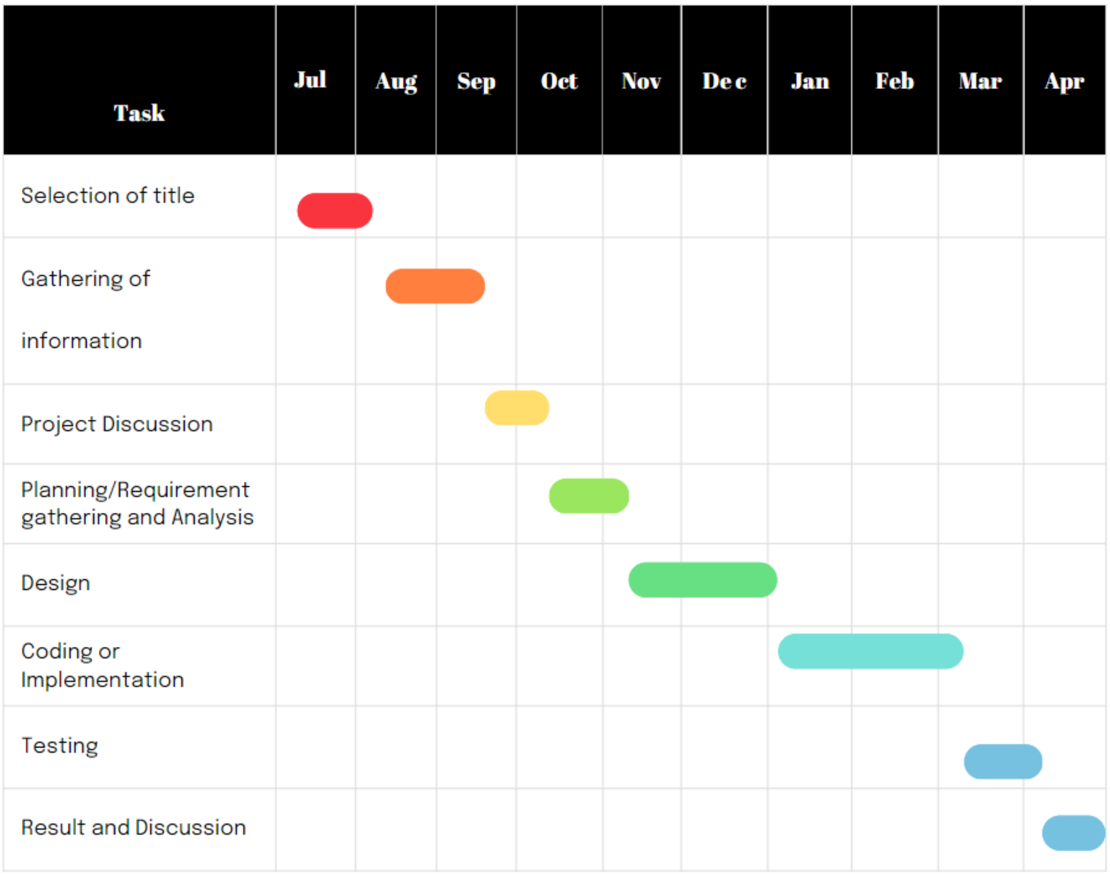


Figure 2.1: Gantt Chart

2.4 Effort Allocation

Effort Allocation is necessary so every team member can give its best to the project. Project was divided into smaller module and task form, for simplification and easy understanding of project overall. Some modules include every team associate's presence to take advantage of team decision taking skills, and some task include some individual member to work on it with precision. We divided the project into 6 modules.

- 1. Gathering of Information
- 2. Planning/Requirement Analysis
- 3. Selection of Life Cycle
- 4. Planning and Management
- 5. Analysis Design UML

- 6. Coding Implementation
- 7. Testing
- 8. Results Discussion

	Lina	Veena	Harshada
Gathering of information	✓	✓	✓
planning / requirement analysis		✓	
selection of life cycle	✓		
planning and management	✓		✓
analysis and design			✓
Coding and Implementation	✓		
Testing	✓		
Results and Discussion	✓	✓	✓

Table 2.2: Effort Allocation

2.5 Cost Estimation

The basic COCOMO estimation model is given by the following expressions:

Effort = a1 x $(KLOC)^{a2}$ PM .... Eq.(1)

Tdev = b1 x $(EFFORT)^{b2}$ Months .... Eq.(2)

Productivity = $\frac{KLOC}{PM}$ .... Eq.(3)

Where, KLOC is the estimated size of the software product indicated in a1,a2,b1, and b2 are constants for each group of software products, T devis the estimated time to develop the software, expressed in months, Effort is the total effort required to develop the software product, expressed in person-months (PMs) in Table 2.3.

From eq.1 calculate the Effort required for software production.

Effort = a1 x $(KLOC)^{a2}$ PM

= 2.5 x $(1323)^{1.05}$

= 4737.81 PM

From eq.2 Calculate the estimated time to develop Software.

$$\text{Tdev} = \text{b1 x } (Effort)^{b2} \text{ PM}$$
$$= 2.5 \text{ x } (4737.81)^{0.38}$$
$$= 62.33 \text{ Months}$$

From eq.3 Calculate the Productivity of the Project.

$$\text{Productivity} = \frac{KLOC}{PM}$$
$$= \frac{1323}{4737.81}$$
$$\text{P} = 0.2793 \text{ KLOC/PM}$$

Table 2.3 Cost Estimation

1	Total number of persons working on the project	3 Person
2	Time Taken (in months)	4 Months
3	Total Time Allocated per Day (in terms of hrs)	3 hrs
4	Actual Working Hours	430 hrs
5	Cost per hour	Rs 40
6	Total Estimated Project Cost for a Person	Rs 38,400
7	Total Estimated Project Cost For 3 Person	Rs 1,15,200
8	Total Estimated Project Cost For Total Project	Rs 1,53,600

2.6 Summary

The project, is hence found to be feasible since there is a balance of resources required and the cost incurred. Also the project helps the government to identify crime like child labour. The project will be able to easily integrate with other required systems.

Chapter 3

Analysis

The development of computer-based information system includes the system analysis phase which produces or enhances the data model which itself is to creating or enhancing a database. There are a number of different approaches to system analysis. The analysis is the process which is used to analyze, refine and scrutinize the gathered information of entities in order to make consistence and unambiguous information. Analysis activity provides a graphical view of the entire System. System Analysis is the process of gathering and interpreting facts, diagnosing problems and using the facts to improve the system. System analysis chapter will show overall system analysis of the concept, description of the system, meaning of the system. System analysis is the study of sets of interacting entities, including computer system analysis.

The organization of this Chapter is as follows. Section 3.1 represents Requirement Collection and Identification. Software Requirement and Specification are described in the Section 3.2. Section 3.3 describes summary of the chapter.

3.1 Requirement Collection and Identification

Requirement collection is the process which is used to gather, analyze, and documentation and reviews the requirements. Requirements describe what the system will do in place of how. In practical application, most projects will involve some combination of these various methods in order to collect a full set of useful requirements. Requirements collection is initiated when the project need is first identified and the project "solution" is to be proposed. Requirements refinement continues after the project is "selected" and as the scope is defined, aligned and approved.

The system will require only images of the monument to be analysed which will be uploaded by user either through the web-application or CLI.

3.2 Software Requirements Specifications (SRS)

Software Specification will provide a broad understanding of the requirement specification of this system. Also, understand features of this system along with the requirements. Software Requirement Specification documents guide the developers in the development process and it will help to reduce the ambiguity of the requirements provided by the end-user. It's used to provide critical information to multiple teams — development, quality assurance, operations, and maintenance. This keeps everyone on the same page.

3.2.1 Product Feature

The product features are high level attributes of a software or product such as software performance, user-friendly interface, security portability, etc. These attributes are defined according to the product, in this case, a software product.
They are as follows:

- The user will just need to upload image of monument
- The user will be able to view the results of submitted images.
- Our websites enables to view full information of predicted monuments.
- The user will be provided with location of monuments

3.2.2 Operating Environment

The software will operate within the following environment:

- Operating System: Windows 10 or later/Linux/MacOS
- python environment required.
- Any system with at least 4GB RAM
- System with processor Intel CORE i3 or later

3.2.3 Assumption

- It is assumed that the web portal will load and render correctly and as expected on the operating machine.
- It is assumed that the user will have a working internet connection with sufficient internet speed.

- It is assumed that the user is able to either upload images through web interface or CLI.
- It is assumed that the user will upload the images within the given specifications.

3.2.4 Functional Requirements

Functional requirements are the functions which are expected from the software or platform. Functional requirements along with requirement analysis help identify missing requirements. They help clearly define the expected system service and behavior.
Functional requirements are as follows:

- To be able to upload the images through the web interface and command line interface.
- To be able to view the results of analysis through web as well as command line interface.

3.2.5 Non-Functional Requirements

Non-functional Requirement is mostly quality requirement. That stipulates how well the portal does, what it has to do. Other than functional requirements in practice, this would entail detail analysis of issues such as availability, security, usability and maintainability.
Non-functional requirements are as follows:

- The processing system is fast enough to reduce the existing delays.
- The interface should be simple and minimal.
- The results should be comprehensive and detailed.

3.2.6 External Interfaces

■ User interface

The proposed system has several options for users to interact with. Following are the user interfaces available:

- Web-application (GUI)
- Command Line Interface (Terminal based interface)
  The web application will be available so that the users will be able to upload images through a simple GUI and the CLI will be available so that the user will be able to integrate it with other systems easily.

- Software interface

  The only software interface required for this project is the Application Programming Interface with the Jupyter Notebook which will then process the images. This software interface will run on local server along with the Jupyter server.

3.3 Summary

In the chapter, Analysis was presented which included the hardware and software requirements, functional and non-functional requirements and the software requirements specification(SRS) as well. In the next chapter, Design is described along with various UML diagrams.

Chapter 4

Design

Design is the activity to design and model the various component of software system. The system design provides the understanding and procedural details necessary for implementing the system. Design is helpful for a better understanding of the project. It contains the UML diagrams, data flow diagrams. UML is a modeling language which is used to document the object-oriented analysis and design.

The organization of this Chapter is as follows. Section 4.1 describes the system architecture of the project. DFD of the project are represented in Section 4.2. Section 4.3 represents UML Diagrams (Use case, Class, Sequence, Component, Deployment, State chart, Activity diagram, Class Diagram, Component Diagram, etc.) of the project. Finally, the Summary is described in last Section 4.4

4.1 Algorithm

In this project we are using Convolutional Neural Networks (CNNs) Algorithm. CNNs are successful for tasks like object recognition, picture classification, and image segmentation since they are built to automatically learn characteristics from images. Convolution involves applying a set of learnable filters (also known as convolutional kernels) to the input image, which results in feature maps that highlight patterns in the image. Here, the convolutional layers can learn to detect edges, corners, textures, and other local features. Non-linearity is important for recognizing the subtle variations and nuances in images, such as differences in lighting, scale, and viewpoint, which are often present in monument images taken from different angles or under different lighting conditions [8]. To add non-linearity to the model, CNNs often include activation functions (such as ReLU, sigmoid, or tanh).

To train a CNN for monument recognition and classification, a large dataset of labeled monument images is typically used to optimize the model's parameters through a process called back propagation. During training, the model learns to automatically adjust the weights of the convolutional kernels and dense layers to minimize the error between its predicted outputs and the ground truth labels [10]. The model's performance is evaluated using metrics such as accuracy, precision and it is iteratively fine-tuned until satisfactory performance is achieved.

In conclusion, CNNs are powerful deep learning models that can automatically learn and extract relevant features from images, making them highly effective for image processing tasks. By leveraging convolution, non-linearity, pooling, feature fusion, and classification, CNNs can capture local and global contextual information from images, enabling them to recognize and classify monuments.

4.2 System Architecture

Systems Architecture is a generic discipline to handle objects (existing or to be created) called "systems", in a way that supports reasoning about the structural properties of these objects. The system architecture is the conceptual model that defines the structure, behavior and more views of a system.

An architecture description is a formal description and representation of a system. It provides broad understanding of the portal. In the system architecture database provide the functionality like get information, select criteria, etc. to users.

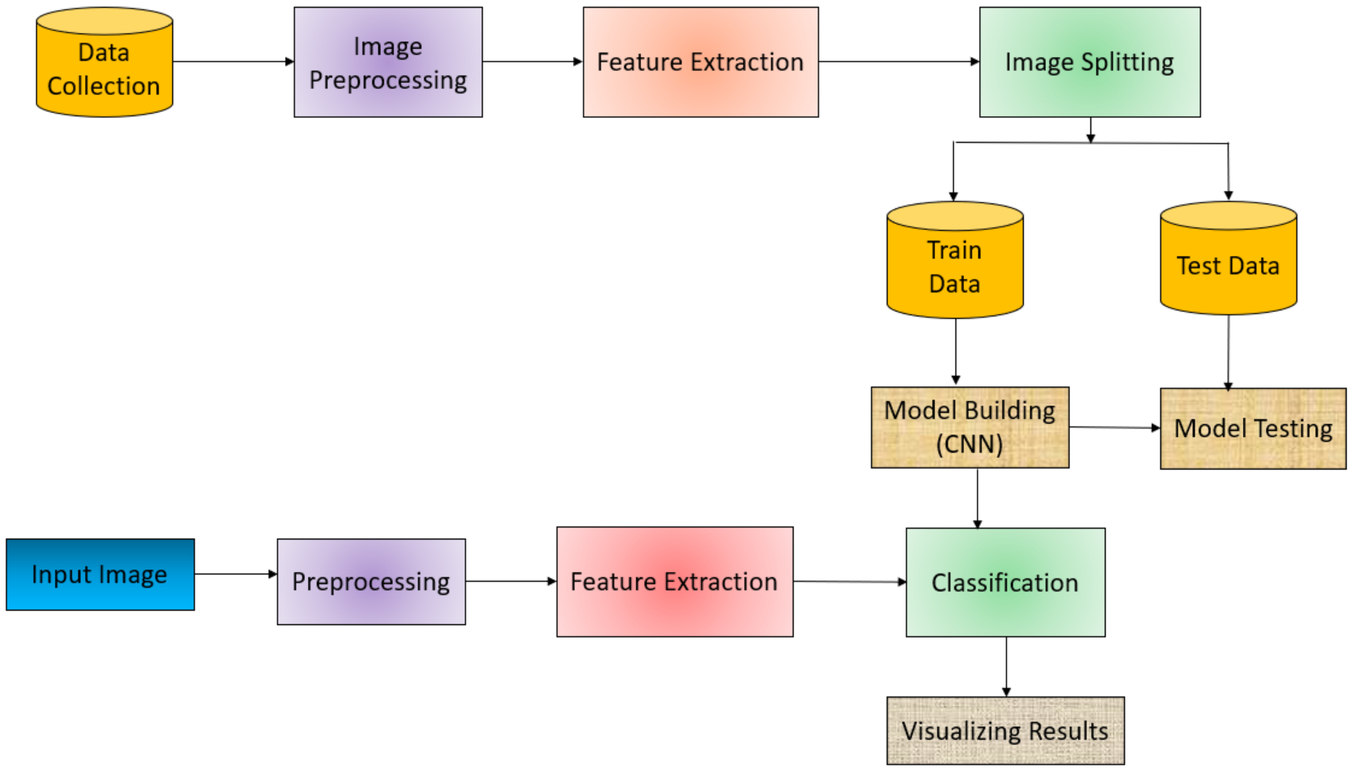


Figure 4.1: System Architecture

4.3 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the ‘flow’ of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step 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 kind of information 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.

4.3.1 Level 0 DFD

Level 0 contains one input and one output. The system provides information to the user means system is input and the user is output. Figure 4.2 shows Level 0 DFD of project.

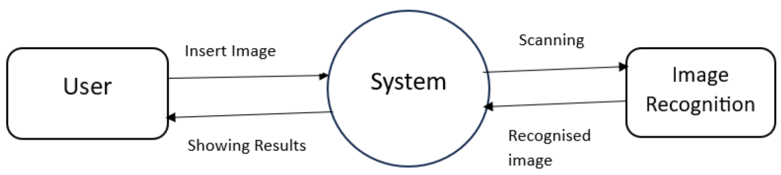


Figure 4.2: 0 Level Data Flow Diagram

4.3.2 Level 1 DFD

A level 1 DFD notates each of the main sub-processes that together form the complete system. We can think of a level 1 DFD as an "exploded view" of the context diagram. Figure 4.3 shows Level 1 DFD of project.

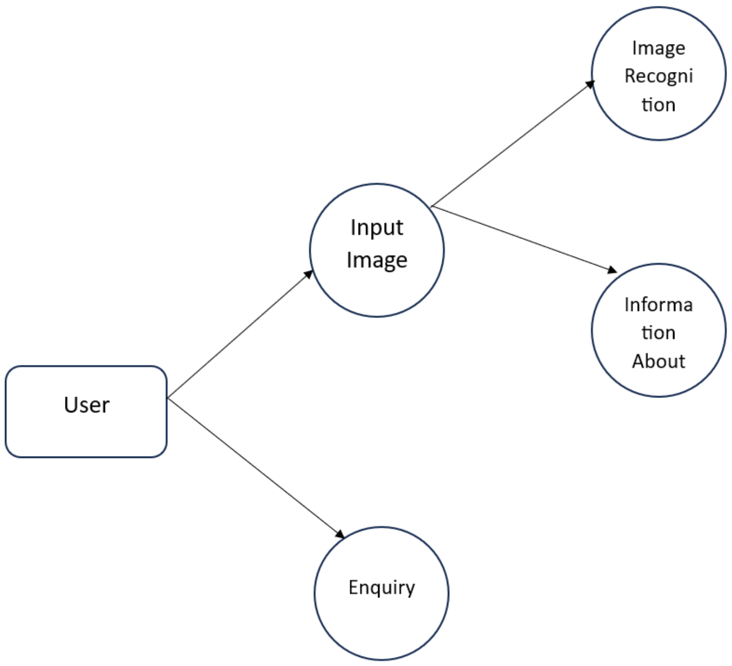


Figure 4.3: 1 Level Data Flow Diagram

4.3.3 Level 2 DFD

A level 2 data flow diagram offers a more detailed look at the processes that make up an information system than a level 1 DFD does. It can be used to plan or record the specific makeup of a system.
Figure 4.4 shows Level 2 DFD of project.

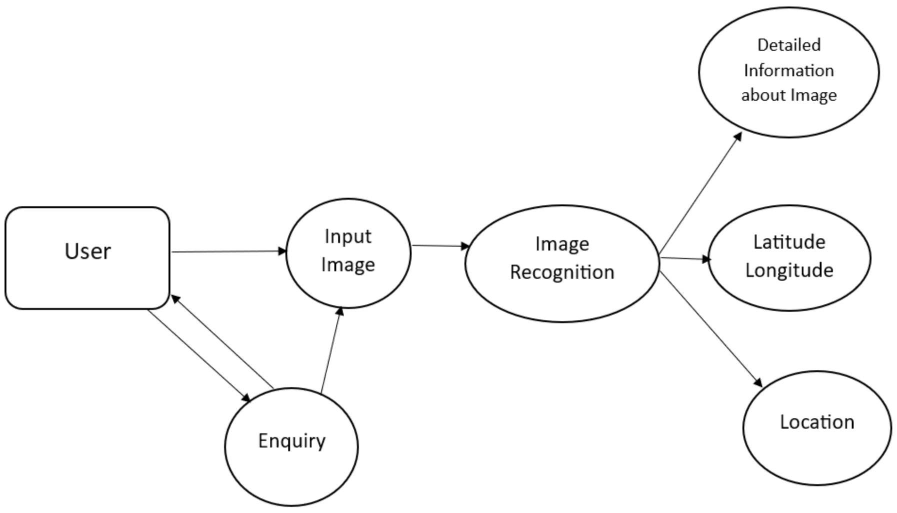


Figure 4.4: 2 Level Data Flow Diagram

4.4 UML Diagrams

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

4.4.1 Use Case Diagram

Use case diagram shows the interaction between Use case which represents system functionality and actor which represent the people or system. Fig 4.5 shows use case diagram.

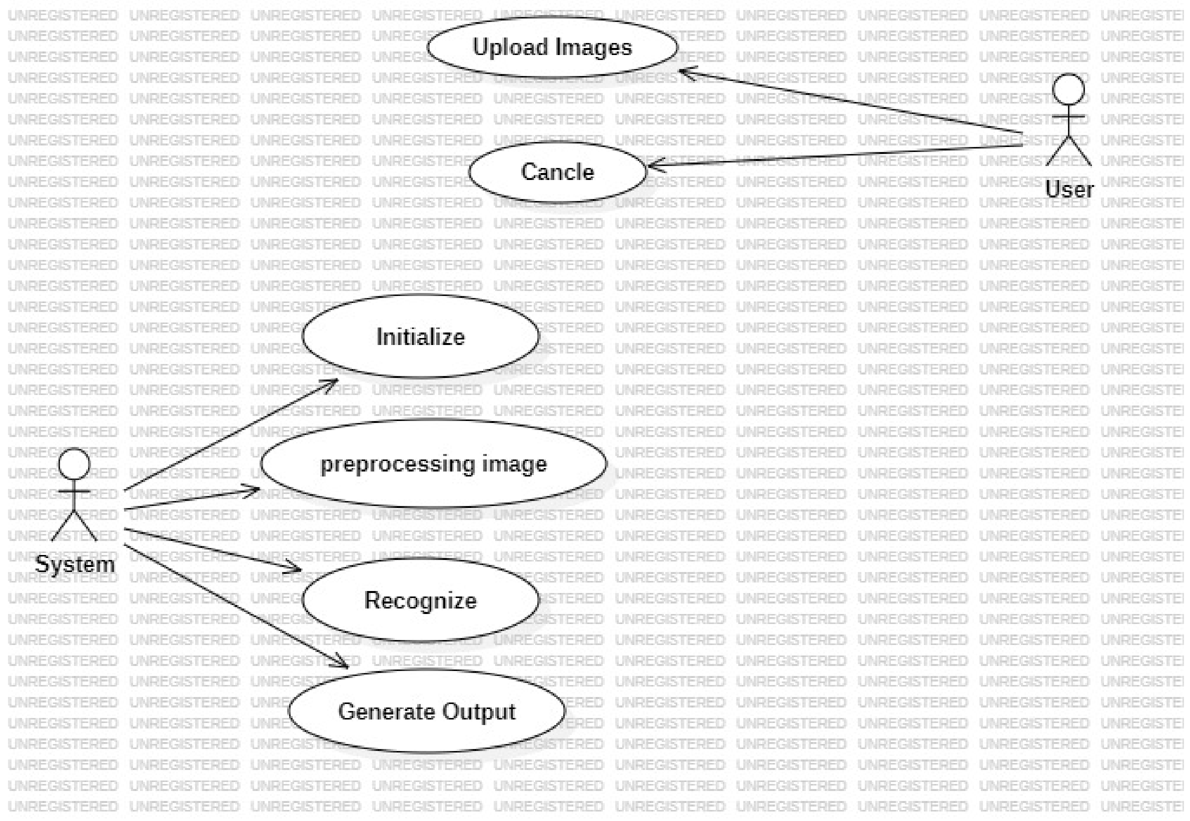


Figure 4.5: Use Case Diagram

4.4.2 Sequence Diagram

The sequence diagram shows the flow of functionality through Use case. A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.Fig 4.6 shows sequence diagram.

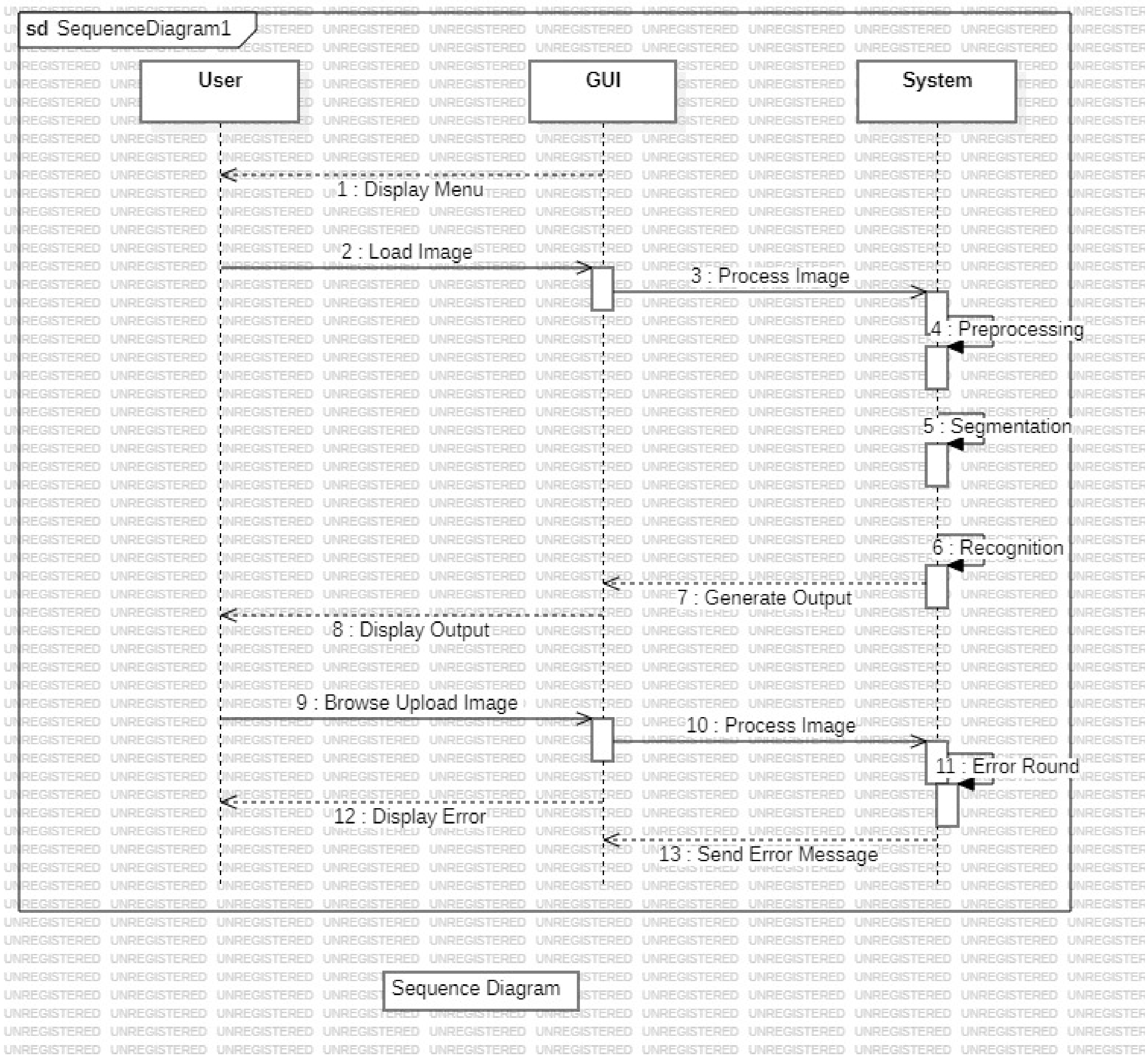


Figure 4.6: Sequence Diagram

4.4.3 Collaboration Diagram

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object. Fig 4.7 shows collaboration diagram.

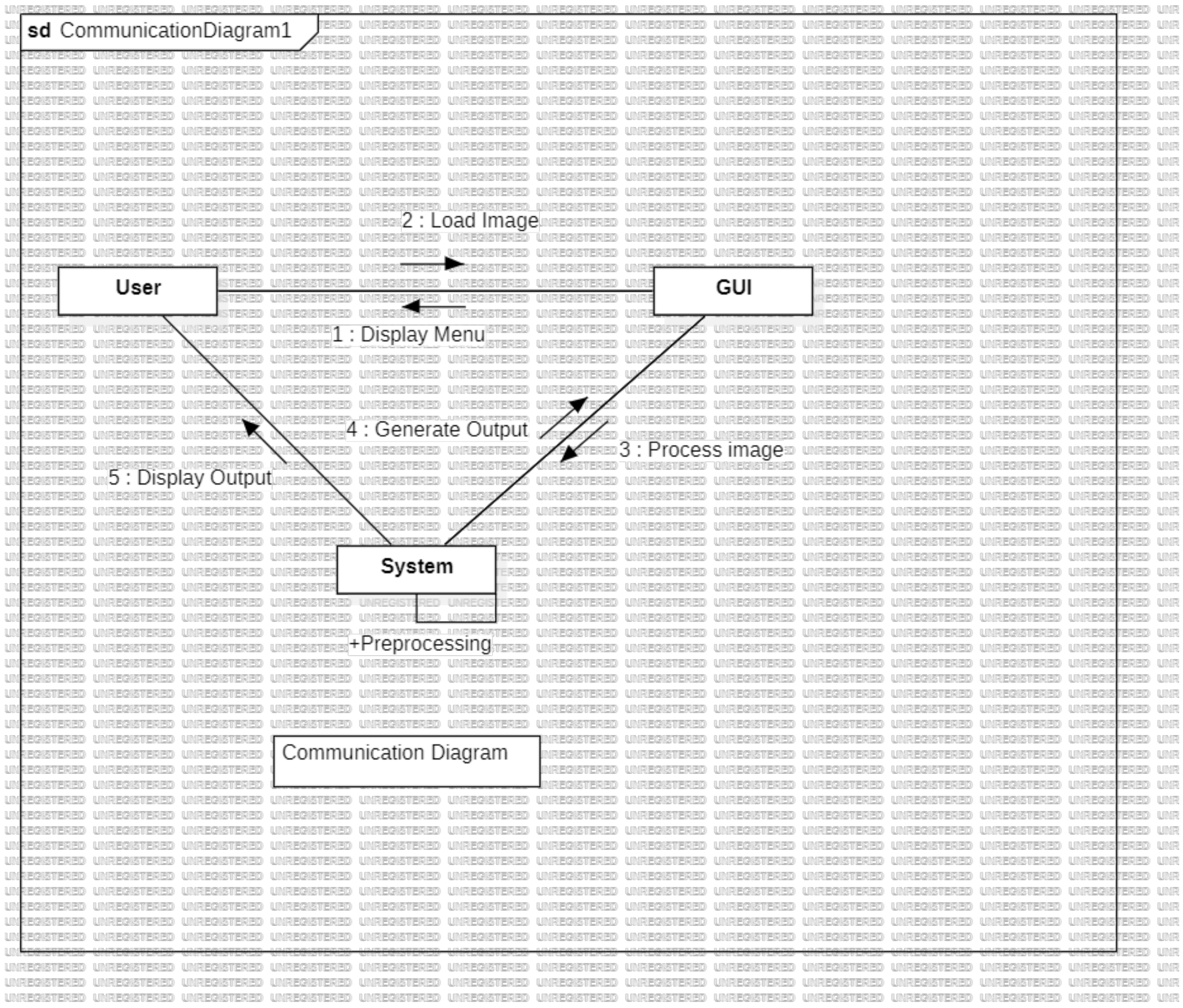


Figure 4.7: Collaboration Diagram

4.4.4 Class Diagram

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. Fig 4.8 shows class diagram.

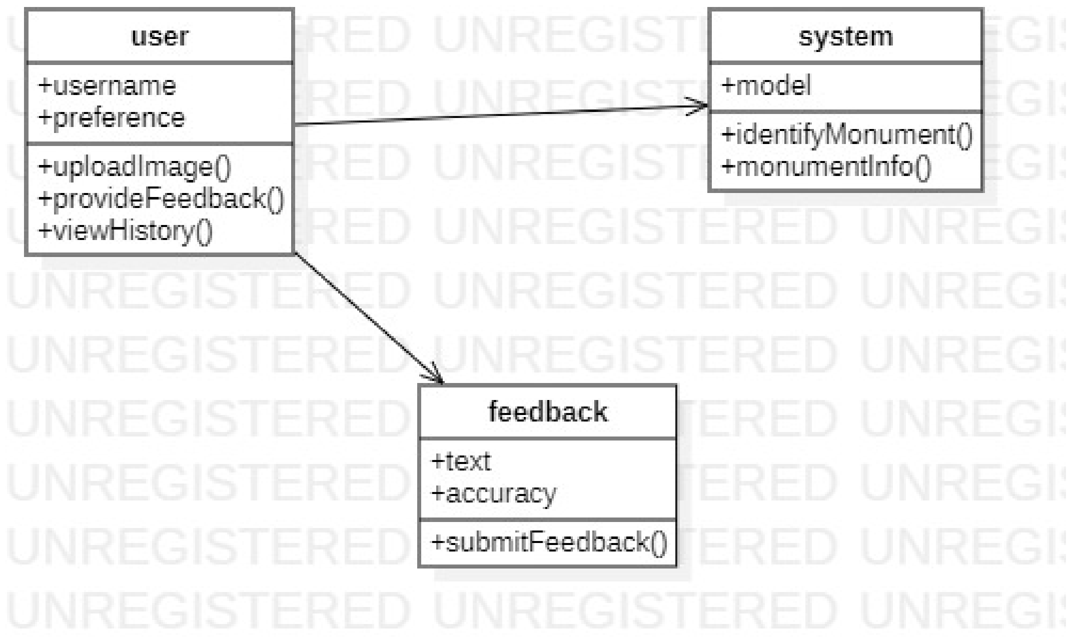


Figure 4.8: Class Diagram

4.4.5 Component Diagram

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development.Fig 4.9 shows component diagram.

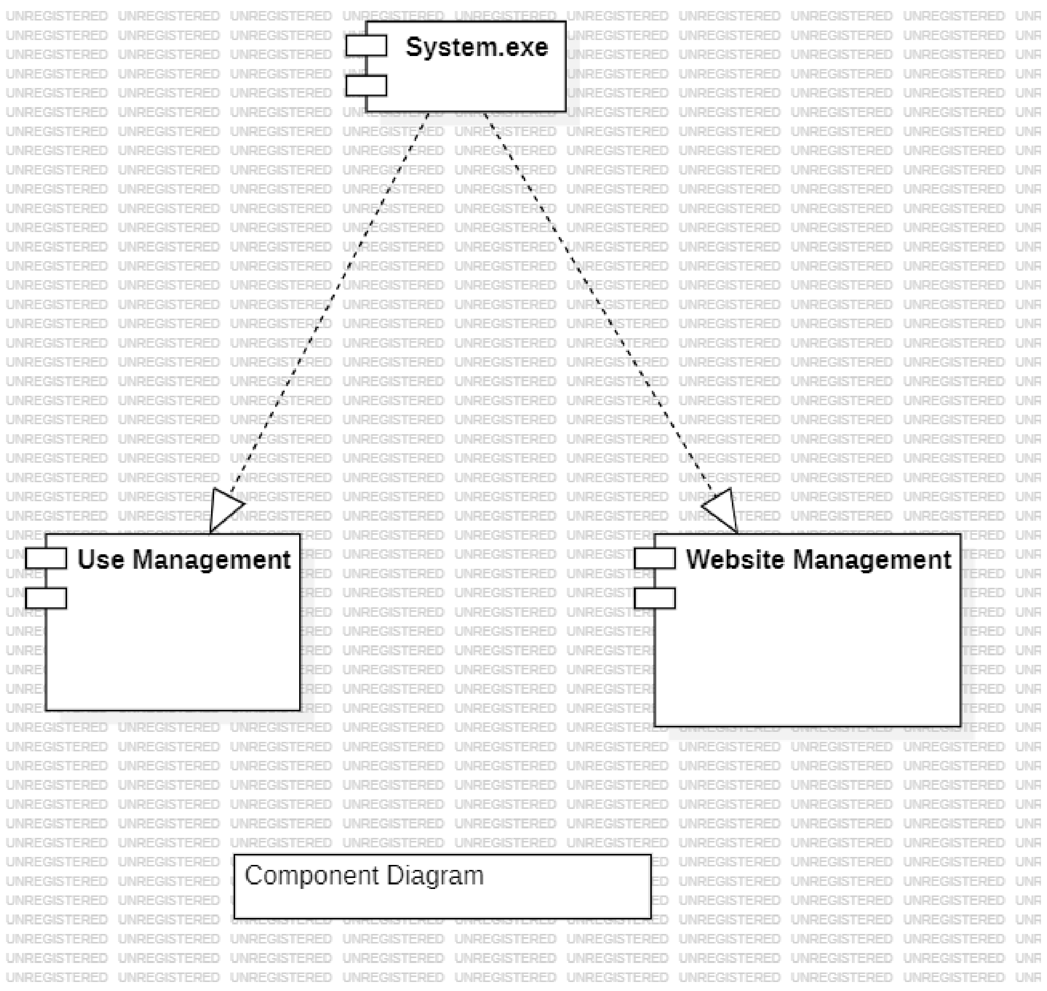


Figure 4.9: Component Diagram

4.4.6 Deployment Diagram

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middle ware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system. Fig 4.10 shows deployment diagram.

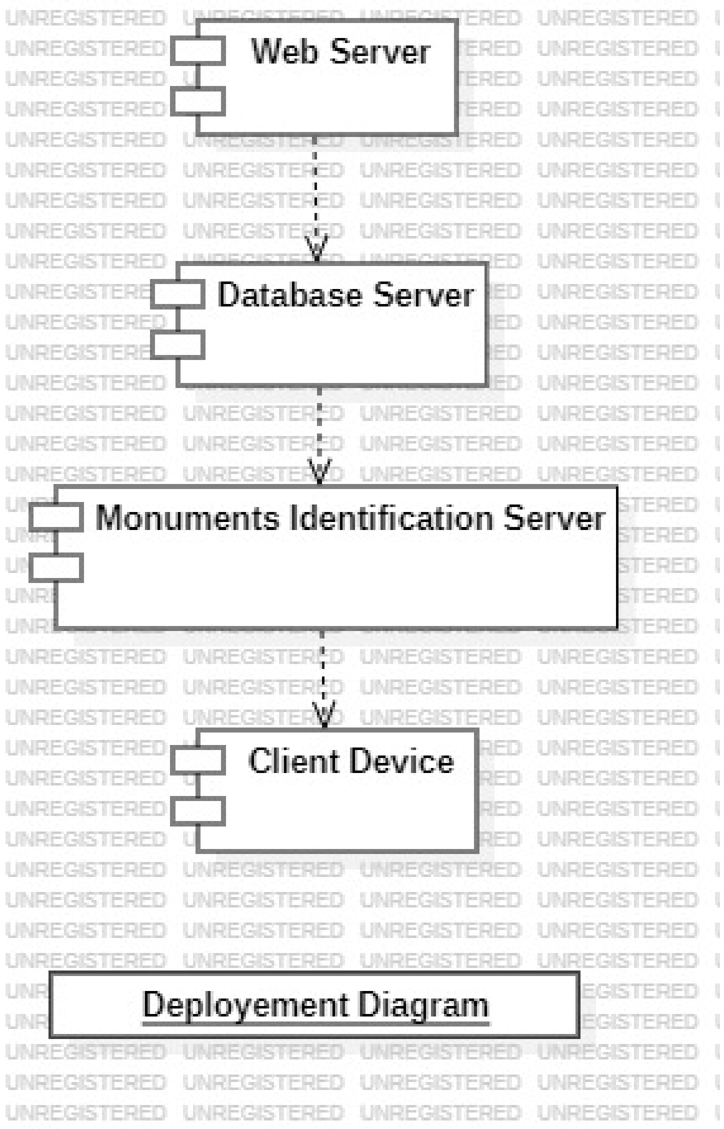


Figure 4.10: Deployment Diagram

4.5 Summary

Detailed design of project has been described in this chapter including the Data Flow Diagrams and the UML Diagram explaining all the design details of the project. Conclusion of the project has been explained in the next chapter.

Chapter 5

Coding/Implementation

In the Implementation of a Heritage Identification of Monuments involves several steps, including data collection, preprocessing, model selection, training, Testing, and Visualization of Results.

The organization of the chapter is as follows: section 5.2 Algorithm is present section 5.3 software and hardware Requirement is present Section 5.3 presents Modules in Project and section 5.4 Present Summary.

5.1 Algorithm

Following some steps of the algorithm as follows.

1. Importing some useful libraries.
2. Load Model and Labels.
3. Define Image Processing Function.
4. Define Geocoding Function.
5. Main Function(run).
6. Run the Application.
7. Visualise the results.

5.2 Software and Hardware Requirement

Following some requirements of hardware and software.

- Windows 10 or higher.
- Dual Core Processor .
- 160 GB Hard disk .
- 4GB RAM(minimum).
- CPU 2.0 MHZ(or faster).
- Python .
- Google Colab
- Jupyter
- Deep Learning Libraries

5.3 Modules in Project

The following modules are used in the project.

- Streamlit.
- PIL (Python Image Library).
- Tensorflow.
- tensorflowhub .
- numpy.
- Pandas.
- geopy.
- random.

5.4 Summary

In this Chapter, the Implementation part is discussed. In the next Chapter, Testing is presented.

Chapter 6

Testing

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements. Software Testing is a process of verifying and validating whether the program is performing correctly with no bugs. It is the process of analyzing or operating software for the purpose of finding bugs. It also helps to identify the defects / aws / errors that may appear in the application code, which needs to be fixed. Testing not only means fixing the bug in the code, but also to check whether the program is behaving according to the given specifications and testing strategies. This Chapter is organized as follows.

Section 6.1 describes Black Box Testing and white Box Testing. Section 6.2 describes Manual Testing and Automated Testing. Test Cases Identification and Execution describe in Section 6.3. Finally, summary of the chapter is given in last section

6.1 Black Box and White Box Testing

The following methodologies are used for testing.

6.1.1 Black Box Testing

Black Box testing also known as Behavioral Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional. This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories

- Incorrect or missing functions
- Interface errors

- Errors in data structures or external database access
- Behaviour or performance errors
- Initialization and termination errors

6.1.2 White Box Testing

White Box Testing is also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing. It is a software testing method in which the internal structure, design, implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system. This method is named so because the software program, in the eyes of the tester, is like a white or transparent box; inside which one clearly sees.

6.2 Manual and Automated Testing

Following methodologies are used for testing.

6.2.1 Manual Testing

It is the oldest and most rigorous types of testing it is performed by human sitting in front of a computer carefully going through applications screens, trying various usage and input combinations, comparing the results to the expected behavior and recording their observations about project. There are certain ways of manual testing first of all test cases are written then they are executed and then report is generated according to test result.

6.2.2 Automated Testing

Automation testing is a Software testing technique to test and compare the actual outcome with the expected outcome. This can be achieved by writing test scripts or using any automation testing tool. Test automation is used to automate repetitive tasks and other testing tasks which are difficult to perform manually. The benefit of manual testing is that it allows a human mind to draw insights from a test that might otherwise be missed by an automated testing program. Automated testing is well-suited for large projects; projects that require testing the same areas over and over; and projects that have already been through an initial manual testing process.

6.3 Test Case Identification and Execution

6.3.1 Test Cases

Test Case is the set of inputs along with the expected output and actual output some additional information.

ID	Scenario	Input	Expected Output	Actual output	Result
1	Information without any fabrication	Image/text	No fabrication or false information detected	No fabrication or false information detected	Pass
2	Information with fabrication	Image/text	Fabrication detected	Fabrication detected	Pass
3	Information with fabrication	Image/text	Fabrication detected	Fabrication detected	Pass
4	Valid information gathered	Image/text	Information is directed to authorities	Information is directed to authorities	Pass
5	Valid information gathered	Image/text	Information is directed to authorities	Information is directed to authorities	Pass

Chapter 7

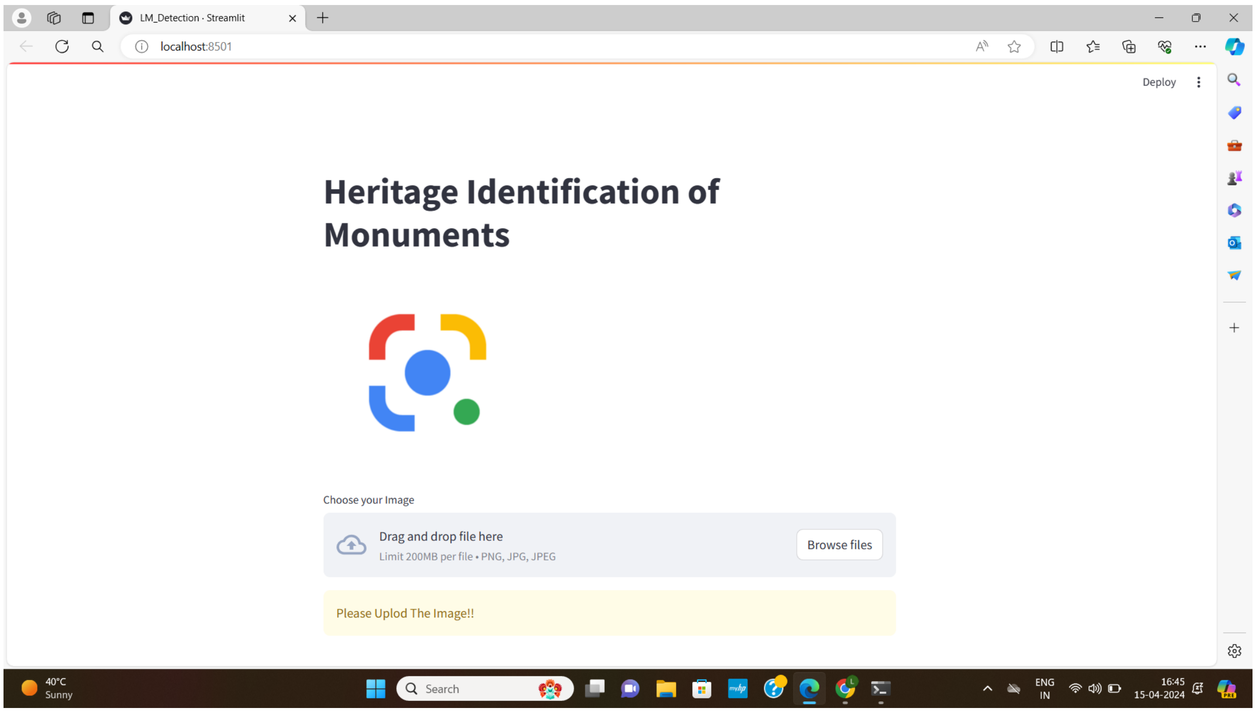
Results and Discussion

This chapter discusses about the results of the project. The application successfully utilizes a pre-trained model for landmark classification specific to Asia. Users can upload images, and the app predicts the landmark depicted in the image. For the predicted landmark, the application retrieves its address, latitude, and longitude using geocoding. The application provides a user interface with functionalities for: Uploading images, Displaying predicted landmark name, Displaying the uploaded image, Displaying address (if geocoding successful), Showing latitude and longitude (optional buttons), Opening the location on Google Maps (using hyperlink).

Easy to use: The application offers a user-friendly interface for image upload and landmark prediction.

Pre-trained model: Utilizing a pre-trained model saves development time and leverages existing expertise in landmark classification.

Geocoding integration: The ability to retrieve location information for the predicted landmark adds value by providing context, It can also be integrated into web as well as mobile apps.



Heritage Identification of Monuments

Choose your Image

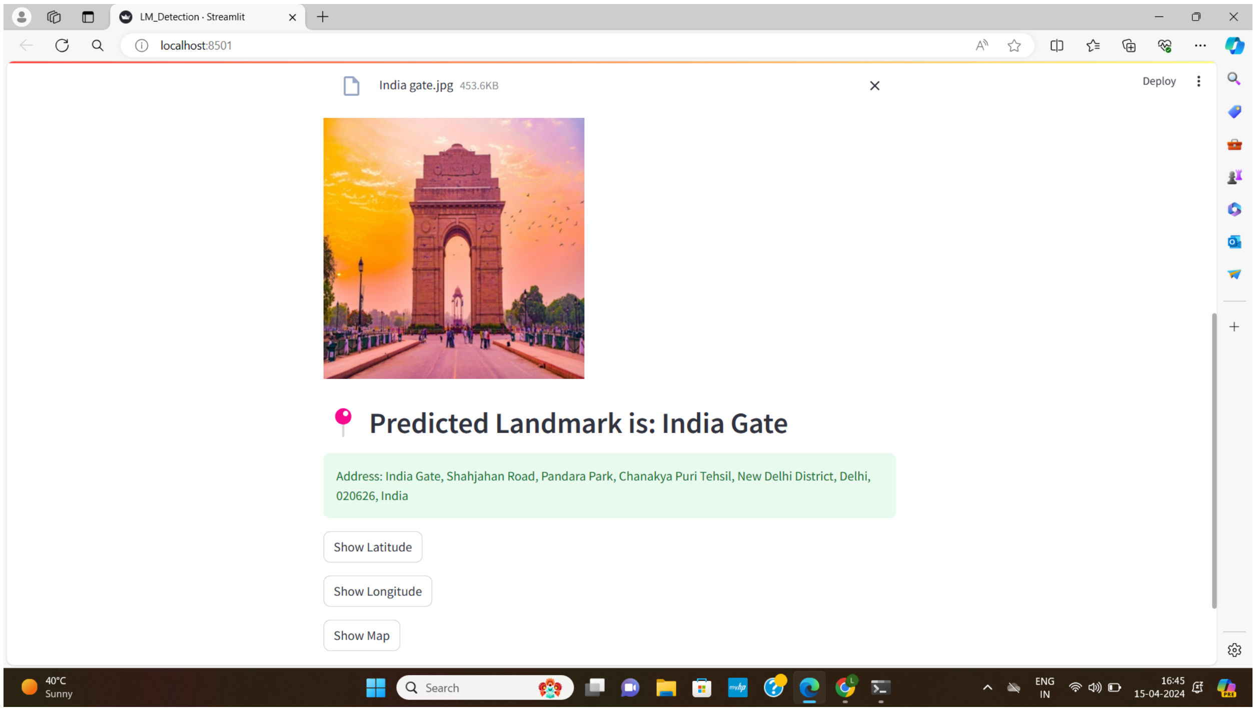
Drag and drop file here
Limit 200MB per file • PNG, JPG, JPEG

Browse files

Please Uplod The Image!!

Figure.7.1 Input image

Here, User can see first page of our system. where, system is labelled with its title as Heritage Identification of Monuments with logo and can see option as Drag and image file here. and if user is not able to upload image then giving exception message as Please upload the image.



India gate.jpg 453.6KB

Predicted Landmark is: India Gate

Address: India Gate, Shahjahan Road, Pandara Park, Chanakya Puri Tehsil, New Delhi District, Delhi, 020626, India

Show Latitude

Show Longitude

Show Map

Figure.7.2 Predicted output

Once the image is uploaded, the application processes it to predict the landmark it depicts.

The uploaded image is displayed in the application, allowing users to visually confirm the landmark they have uploaded. This feature helps users ensure that the correct image has been uploaded before proceeding with the prediction and other interactive features provided by the application.

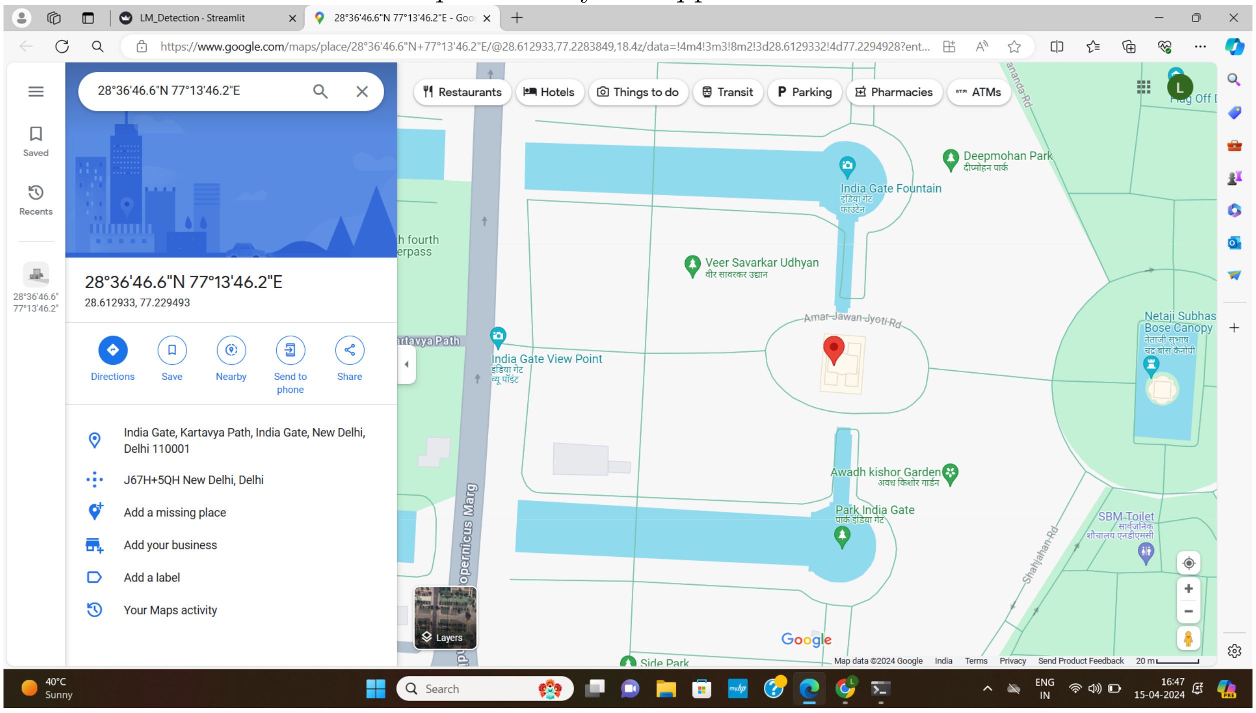
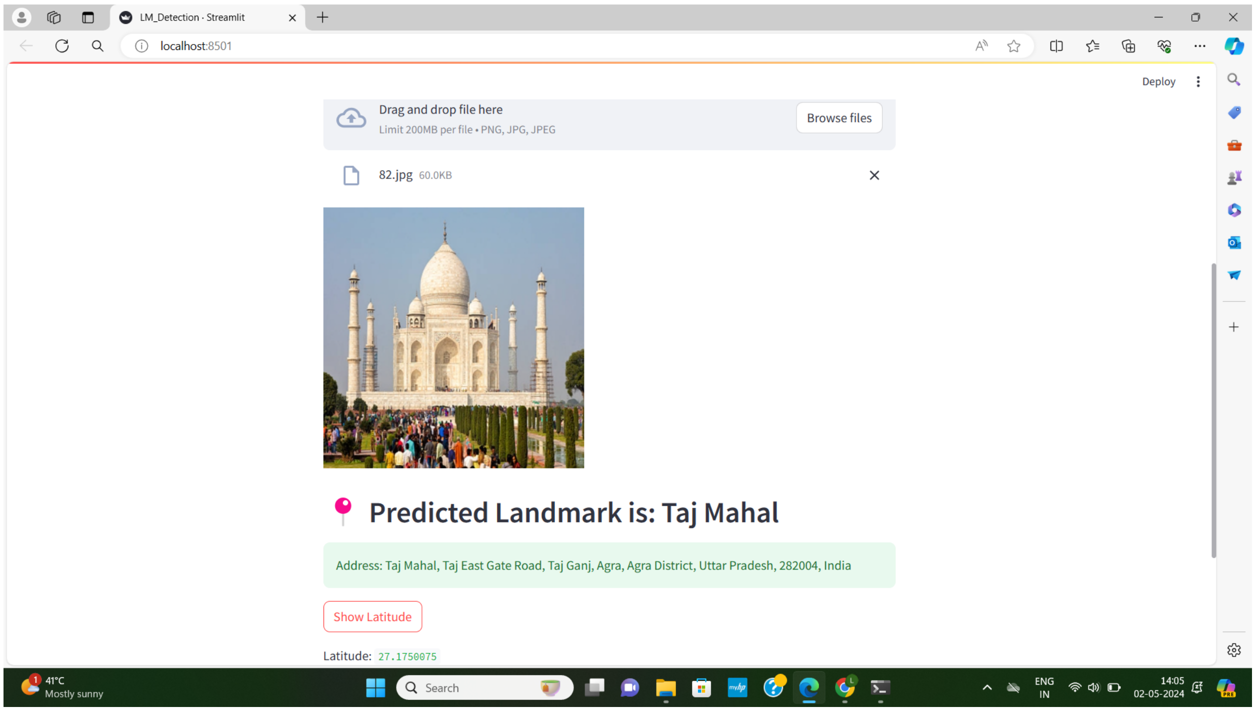


Figure.7.3 Map of predicted of image

When a user clicks the "Show Map" button in our Streamlit web application, the system generates a link to Google Maps that displays the location of the predicted landmark. This link opens in a new tab or window, showing a map centered on the latitude and longitude coordinates of the predicted landmark. Users can interact with the map, zoom in or out, and explore the surrounding area to get a better understanding of where the landmark is located. This feature provides a visual representation of the predicted landmark's location, enhancing the user experience and helping them better understand the context of the landmark.



Predicted Landmark is: Taj Mahal

Address: Taj Mahal, Taj East Gate Road, Taj Ganj, Agra, Agra District, Uttar Pradesh, 282004, India

Latitude: 27.1750075

Figure.7.1 Taj Mahal Image prediction

Here we are uploding another image of Monumenet ie. of Taj Mahal.

The identification and preservation of our cultural legacy could be revolutionised by the use of deep learning in heritage identification, which is a promising area of research. Our heritage monuments can be preserved for future generations with the aid of cutting-edge technologies and deep learning algorithms. However, there are still a number of issues that must be resolved, including the need for more diverse datasets and a lack of adequate documentation.

Chapter 8

Conclusion and Future Work

We successfully developed a system to recognize and identify the heritage of the monument. A dataset of satellite imagery of monuments served as the foundation for the development of a deep learning algorithm called Convolutional Neural Network. To do this, we first preprocessed the images in the dataset and then extracted the features from those preprocessed photos along with splitting of dataset, building and evaluating the model. Finally, the findings demonstrate the accuracy and ability to forecast the type of monuments and whether they are patrimony or not. Overall, this effort emphasises how critical it is to use contemporary technologies to preserve our cultural history.

Future Improvements:

Model exploration: Explore other pre-trained models or fine-tune the existing model for potentially higher accuracy or broader landmark coverage.

User interface enhancements: Incorporate layout organization using Streamlit containers and columns, add visual elements like progress bars for image processing, and improve error handling with informative messages.

Advanced features: Implement functionalities like landmark information display (historical details, description), user account creation for model training/customization based on user data (if applicable), or image pre-processing options (cropping, resizing).

Chapter 9

Bibliography

- Official UNESCO site for Criteria of selecting a site for heritage declaration (https://whc.unesco.org/en/criteria/)
- Book: Deep Learning for Computer Vision by Rajalingappaa Shanmugamani, 2018
- A. Crudge, W. Thomas and K. Zhu, "Landmark Recognition Using Machine Learning," CS229, Project 2017.
- https://www.researchgate.net/publication/371964102HeritageIdentificationofMonuments usingDeepLearningTechniques/link/649fcde3b9ed6874a5eb6c88/downloadtp=eyJjb250ZXh0I jp7ImZpcnN0UGFnZSI6InB1Ymxp Y2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19