"CognisSAR"

A Project Report Submitted to

Rajiv Gandhi Proudyogiki Vishwavidyalaya

Towards Partial Fulfillment for the Award of

Bachelor of Technology
in
Computer Science & Information Technology

Submitted by:	**Guided by:**
Chetna Sontaki (0827CI211051)	**Guide Name: Prof. Nidhi Nigam**
Dhruv Sharma (0827CI211061)	**Designation**
Divyanshu Aaliwal (0827CI211067)	**CSIT Department**

Acropolis Institute of Technology & Research, Indore

Jan – June 2025

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

2021-2025

DECLARATION

I hereby declare that the work, which is being presented in this project entitled "**CognisSAR**" in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology in Computer Science and Information Technology**, is authentic record of work carried out by me.

Place: CSIT, Indore **Signature of Student**

Date:

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

2021-2025

RECOMMENDATION

This is to certify that the work embodied in this project entitled **"CognisSAR"** submitted by **Chetna Sontaki (0827CI211051) , Divyanshu Aaliwal (0827CI211067), Dhruv Sharma (0827CI211061)**, is a satisfactory account of the bonafide work done under the supervision of **Prof. Shurti Lashkari, Prof. Nidhi NIgam**, is recommended towards partial fulfillment for the award of the Bachelor of Technology in Computer Science & Information Technology degree by Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal.

Project Guide	**Project Coordinator**
Prof. Nidhi Nigam	**Prof. Shurti Lashkari**

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

2021-2025

CERTIFICATE

The Project entitled **"CognisSAR"** submitted by **Chetna Sontaki (0827CI211051),Divyanshu Aaliwal (0827CI211067), Dhruv Sharma (0827CI211061)**, has been examined and is hereby approved towards partial fulfillment for the award of **Bachelor of Technology** in **Computer Science & Information Technology,** for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

Internal Examiner	**External Examiner**
Date:	**Date:**

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

2021-2025

STUDENT UNDERTAKING

This is to certify that project entitled **“CognisSAR”** has developed by us under the supervision of **Prof. Nidhi Nigam**. The whole responsibility of work done in this project is ours. The sole intension of this work is only for practical learning and research.

We further declare that to the best of our knowledge, this report does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University / Deemed University without proper citation and if the same work found then we are liable for explanation to this.

Student 1	**Student 2**	**Student 3**
Chetna Sontaki	**Dhruv Sharma**	**Divyanshu Aaliwal**
Date:	**Date:**	**Date:**

ACKNOWLEDGEMENT

We thank the almighty Lord for giving me the strength and courage to sail out through the tough and reach on shore safely.

We owe a debt of sincere gratitude, deep sense of reverence and respect to our guide **Prof. Nidhi Nigam, Designation** and mentor **P Prof. Shurti Lashkari**, **Designation**, AITR, Indore for their motivation, sagacious guidance, constant encouragement, vigilant supervision and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks to **Prof.** (**Dr.) Shilpa Bhalerao HOD CSIT**, AITR Indore for her support, suggestion and inspiration for carrying out this project. We would be failing in our duty if do not acknowledge the support and guidance received from **Prof.** (**Dr.) S. C. Sharma, Director**, AITR, Indore whenever needed. We take opportunity to convey my regards to the **Management of Acropolis Institute, Indore** for extending academic and administrative support and providing us all necessary facilities for project to achieve our objectives.

We are grateful to our parent and family members who have always loved and supported us unconditionally.

Student 1
Chetna Sontaki

Enrollment No:
0827CI211051

Student 2
Dhruv Sharma

Enrollment No:
0827CI211061

Student 3
Divyanshu Aaliwal

Enrollment No:
0827CI211067

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviations

Abbr1: SAR – Synthetic Aperture Radar

Abbr2: AI – Artificial Intelligence

Abbr3: DL – Deep Learning

Abbr4: CNN – Convolutional Neural Network

Abbr5: RGB – Red, Green, Blue

Abbr6: GPU – Graphics Processing Unit

Abbr7: PSNR – Peak Signal-to-Noise Ratio

Abbr8: SSIM – Structural Similarity Index Measure

Abbr9: ROI – Region of Interest

Abbr10: GAN – Generative Adversarial Network

Abbr11: HDT – High-Dimensional Tensor

ABSTRACT

Synthetic Aperture Radar (SAR) images, while valuable for remote sensing and surface analysis, are inherently monochromatic, which limits their interpretability and makes it difficult for analysts to distinguish subtle surface features. This limitation poses a significant challenge in extracting meaningful information from SAR data, particularly for applications requiring precise feature identification, such as environmental monitoring, urban planning, and disaster management. To overcome this issue, this report proposes a novel approach for enhancing SAR images through deep learning-based colorization techniques. The primary objectives of this research are to preprocess SAR data efficiently, design and implement a deep learning model capable of accurately colorizing grayscale SAR images, and evaluate the model's performance in improving the interpretability and usability of SAR data. The deep learning model leverages advanced techniques in image processing and neural networks to generate realistic colorized images that provide clearer insights into surface features. The report discusses the model architecture, training process, and evaluation metrics, and it demonstrates the impact of colorization on the interpretability of SAR data. Results show that the proposed solution significantly improves feature detection, aiding in more accurate analysis of SAR data across various domains. This work presents a promising direction for enhancing the application of SAR technology in a wide array of remote sensing fields.

Chapter 1: Introduction

1.1 Overview

Synthetic Aperture Radar (SAR) imaging technology provides a powerful tool for remote sensing by enabling the collection of high-resolution imagery regardless of weather conditions or lighting. However, SAR images are typically captured in monochromatic (grayscale) format, which limits their interpretability by remote sensing analysts who rely on color to distinguish surface features and interpret complex patterns. This limitation poses a challenge, as effective interpretation is crucial in fields like environmental monitoring, urban planning, and disaster response.

The aim of this project is to enhance the usability and interpretability of SAR images by introducing a novel approach for their colorization using Deep Learning techniques. This colorization process is intended to improve the accessibility and analysis of SAR data by remote sensing professionals, enabling clearer visual representation of surface characteristics that might otherwise remain indistinguishable.

1.2 Existing System

SAR images are processed using traditional techniques like histogram equalization and pseudo-coloring to enhance visual interpretability, yet these methods fall short in differentiating surface features due to SAR's monochromatic nature. Although some Deep Learning approaches have been applied for general grayscale image colorization, they primarily focus on natural images and do not effectively handle the unique characteristics of SAR data, such as its specific textures and speckle noise. Furthermore, existing model lack adaptability to SAR features, making it difficult to achieve realistic colorization and accurate interpretation, underscoring the need for a specialized Deep Learning approach tailored to SAR imagery.

1.3 Problem Statement

The inherent monochromatic nature of Synthetic Aperture Radar (SAR) images limits their interpretability, making it difficult for remote sensing analysts to distinguish between different surface features. This lack of color hinders effective analysis, particularly in applications such as environmental monitoring, urban planning, and disaster response. To address this issue, there is a need to develop a novel Deep Learning approach for the colorization of SAR images. The objectives are to preprocess SAR data, design and implement an advanced model capable of accurately adding color to grayscale SAR images, and evaluate the model’s effectiveness in enhancing the usability and interpretability of SAR data for a range of remote sensing applications.

1.4 Proposed System

This project proposes a Deep Learning-based approach to colorize monochromatic SAR images, enhancing interpretability for remote sensing applications. The system will preprocess SAR data, reduce noise, and normalize intensities, followed by implementing a custom neural network model tailored to SAR characteristics. The model will use advanced techniques, like CNNs and GANs, to produce realistic color mappings that represent surface features accurately. The effectiveness of the colorization will be evaluated using quality metrics and expert feedback, ultimately improving SAR image usability in fields like environmental monitoring, urban planning, and disaster response.

1.5 Need and Scope

Need:

The monochromatic nature of Synthetic Aperture Radar (SAR) images limits their interpretability, making it challenging for remote sensing analysts to differentiate between surface features effectively. This constraint hinders the application of SAR data in critical fields such as environmental monitoring, urban planning, and disaster management, where accurate visual representation of surface characteristics is essential. Current colorization techniques lack the specificity required for SAR data, underscoring the need for an advanced Deep Learning-based solution to generate realistic, color-enhanced SAR imagery for improved analysis.

Scope:

This project focuses on developing a specialized Deep Learning model to colorize SAR images, significantly enhancing their usability for diverse applications. The model will be designed to capture the unique textural and reflective properties of SAR data, creating a realistic color representation that aids in feature differentiation. The scope of this project includes SAR data preprocessing, model design, training, and performance evaluation. The colorized images generated by this model will serve as valuable resources for remote sensing professionals across various fields, broadening the practical utility of SAR imagery and enabling more effective decision-making in areas such as natural resource management, land use monitoring, and emergency response.

1.6 Report Organisation

- Chapter 1 states the overview of the project with discussing about the discussing about the existing systems in today’s scenario. Describing about the problem statement we are facing about the system. We have given our proposed solution considering all the shortcoming of the previously used system.

- Chapter 2 states the literature survey i.e. the background details of our system including the software engineering paradigm and explaining about the technologies (Software and Hardware requirement) which we have used building the system.

- Chapter 3 states about the Analysis of the whole system i.e. identification of system requirement about the feasibility study-Technical Feasibility, Financial Feasibility, operational Feasibility.

- Chapter 4 states about the Design of the whole system including all the UML diagrams all the tools used with ER Diagram and Data Flow Diagram and Data Dictionary.

- Chapter 5 states about the whole code of the system, java code , typescript ,html ,css scripts and its integration and adaptability.

- Chapter 6 states the Testing phase of our system all the different testing methods and strategies and we have run test cases.

- Chapter states the conclusion of the whole system explaining the advancement of our project in future.

- References: The books, websites, journals, blogs which we have referred.

Chapter 2: Literature Survey

2.1 Study

1. both organisation user and complainant citizen demands. **Introduction to the Challenge** SAR (Synthetic Aperture Radar) images are widely used in remote sensing applications for mapping and monitoring various surface features. However, the monochromatic nature of SAR images makes it difficult to distinguish and interpret surface features, hindering effective analysis.

2. **Need for Enhancement** The need for enhanced SAR images arises from the challenges faced by analysts in interpreting grayscale SAR data. Colorization of these images can significantly improve the visual differentiation of surface features, thereby facilitating better analysis and decision- making. By utilizing Deep Learning techniques, it becomes possible to automate this enhancement process and provide a solution that reduces manual effort in analyzing SAR imagery.

3. **Current Limitations** Existing methods for SAR image enhancement, including histogram equalization and pseudo-coloring, have proven inadequate for accurate interpretation. These techniques do not adapt to the unique characteristics of SAR data, such as speckle noise and reflectance patterns, resulting in low interpretability and reduced effectiveness in real-world applications.

4. **Proposed Solution** The proposed system seeks to automate and optimize the process of SAR image colorization using a Deep Learning approach. This will reduce the manual effort involved in analyzing SAR data and improve the overall accuracy of surface feature interpretation by generating realistic color representations of SAR images.

5. **Primary Objective** The main objective is to develop a specialized Deep Learning model that can accurately colorize grayscale SAR images, enhancing their usability for remote sensing professionals. This model will address the specific challenges of SAR data, such as speckle noise and low contrast, and improve the interpretability of SAR images.

6. **End Goal** The ultimate goal is to reduce the time spent on manual interpretation of SAR images and eliminate the need for tedious processing. By automating the colorization of SAR data, the system will provide an efficient solution for analysts, allowing them to make better-informed decisions faster, while also improving the overall quality of SAR image interpretation for a wide range of applications.

2.2 Problem Methodology

The Agile methodology for the SAR Image Colorization project follows an iterative and flexible approach. The project begins with initiation, where the scope, stakeholders, environment setup, and a prioritized task backlog are defined. During the sprint planning phase, tasks like data preprocessing, model design, and colorization implementation are broken down into manageable units for execution. The development phase occurs in short sprints (1-2 weeks), where tasks such as preprocessing (noise reduction, normalization), designing the initial CNN model, integrating GANs for improved colorization, and evaluating the model using metrics and expert feedback are completed. Daily standup meetings ensure continuous communication within the team to address challenges and track progress. At the end of each sprint, a review is held to present results and gather feedback, followed by a retrospective to reflect on the sprint's effectiveness and adjust processes. The model is iteratively refined based on feedback, improving both functionality and user interface. Finally, after several sprints, the model is finalized, deployed, and user feedback is collected for ongoing improvements. The Agile approach promotes flexibility, collaboration, and continuous improvement, ensuring the final product meets the project's objectives efficiently.

2.3 Software Engineering Paradigm

The software engineering paradigm for the SAR Image Colorization project follows a structured yet adaptable approach, focusing on the development and deployment of an AI-driven solution. This paradigm combines elements from multiple software engineering methodologies, with an emphasis on model-driven development, iterative refinement, and collaboration between stakeholders. Below is a summary of the chosen paradigm:

1. Model- The SAR Image Colorization project leverages a combination of modern software engineering paradigms to ensure a structured and effective development process. These methodologies prioritize flexibility, continuous improvement, and collaboration, ensuring that the solution meets user needs and is adaptable to changes. Below is an overview of the software engineering paradigms applied in this project:

1. Model-Driven Development (MDD)

Concept: Model-Driven Development focuses on using abstract model to guide the design and development of software applications.

Application in the Project: The SAR colorization process relies on AI model such as Convolutional Neural Networks (CNN) and Generative Adversarial Networks (GAN). These model are designed to learn from SAR data, improving their ability to transform grayscale images into colorized versions. MDD enables the automated design of the colorization process and allows easier adaptation of the model to new requirements or data.

2. Iterative and Incremental Development

Concept: This approach breaks down the development process into small, manageable chunks (iterations), allowing for continuous feedback and improvement.

Application in the Project: The development of the SAR colorization system is conducted in phases. Each sprint focuses on key milestones like data preprocessing, model design, integration of GANs, and model evaluation. The incremental approach ensures that the project is constantly evolving, with regular reviews and iterations based on testing and user feedback.

3. Agile Software Development
Concept: Agile methodology emphasizes flexibility, stakeholder collaboration, and the delivery of software in short, iterative cycles (sprints).
Application in the Project: The SAR image colorization project follows Agile principles, working in short sprints of 1-2 weeks. At the start of each sprint, tasks are defined, and progress is reviewed at the end of each cycle. Continuous feedback from stakeholders (e.g., remote sensing professionals) ensures the development is aligned with user needs, improving the system's usability and functionality over time.
4. Data-Driven Development
Concept: In data-driven development, decisions are made based on data insights rather than predefined rules or assumptions.
Application in the Project: Data is central to the development of the SAR image colorization model. The training data, which includes SAR images, is preprocessed and augmented to improve model performance. The model is iteratively trained and fine-tuned based on the data, allowing it to adapt to the unique characteristics of SAR imagery and improve its ability to generate accurate colorized outputs.
5. Client-Centered Development
Concept: This paradigm places a strong emphasis on understanding and meeting the needs of the end-users (clients).
Application in the Project: Remote sensing professionals and analysts are the primary users of the colorization tool. The development process includes regular consultations and feedback loops with these stakeholders to ensure the tool is user-friendly and meets their specific needs, such as ease of use, image quality, and processing speed.
6. Continuous Integration and Continuous Deployment (CI/CD)
Concept: CI/CD is a software development practice that involves automatically integrating and testing code changes, ensuring that new features or updates are deployed quickly and reliably.
Application in the Project: The SAR image colorization model integrates and tests new changes continuously, ensuring the latest code is always tested and ready for deployment. Automated testing checks for bugs, performance issues, and model accuracy. This approach ensures that updates to the model can be rolled out smoothly without disrupting the system's performance.
7. Quality Assurance and Testing
Concept: Quality assurance involves validating that the software meets the required standards and functions as expected.
Application in the Project: The SAR image colorization model undergoes rigorous testing throughout its development. Quality assurance includes performance evaluations using metrics like Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) to ensure the colorized images are of high quality. User acceptance testing is also conducted to ensure the tool is effective for remote sensing professionals and provides the necessary functionality.

2.4 Software Development Life Cycle:

The Software Development Life Cycle (SDLC) for the SAR Image Colorization project follows a structured approach to deliver a high-quality solution. The main phases include:

Requirement Gathering and Analysis: Understanding the needs of stakeholders, defining the functional and non-functional requirements for the SAR image colorization tool.

System Design: Designing the architecture and model (such as CNN and GAN) for colorization, along with the data processing pipeline.

Implementation (Coding): Writing the code for the colorization model, preprocessing steps, and user interface.

Testing: Conducting unit, integration, performance, and user acceptance testing to ensure the system works as intended and meets user needs.

Deployment: Deploying the system to a production environment, making it accessible to users for colorizing SAR images.

Maintenance and Support: Providing ongoing support, fixing bugs, and improving the system based on user feedback.

Feedback and Iteration: Continuously gathering feedback to refine and enhance the system over time.

2.5 Technology Methodology

2.5.1 Software Requirements

The required software for development environment:

1. Deep Learning Frameworks:

TensorFlow: A powerful open-source framework for machine learning and deep learning. TensorFlow will be used to design and train the convolutional neural network (CNN) and Generative Adversarial Network (GAN) model for image colorization.

Keras: A high-level neural networks API, written in Python and capable of running on top of TensorFlow. Keras simplifies the process of building deep learning model and will be used for developing the architecture of the image colorization model.

2. Image Processing Libraries:

OpenCV: OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library that will be used for image preprocessing tasks such as noise reduction, resizing, and image normalization.

Pillow: Python Imaging Library (PIL) fork, which will be used for image manipulation like conversion, cropping, and filtering of SAR images.

3. Web Development Framework:

React: A popular JavaScript library for building user interfaces. React will be used for developing a dynamic and responsive web-based frontend, allowing users to upload images, view results, and interact with the system easily.

Node.js: A JavaScript runtime environment that will be used for backend development. Node.js will manage user requests, interact with the deep learning model, and handle the communication between the frontend and backend.

Express.js: A fast, unopinionated, and minimalist web framework for Node.js. Express.js will provide a robust set of features to build the backend API for managing requests, serving images, and interacting with the machine learning model.

4. Database Management:

MongoDB: A NoSQL database used for storing and managing image metadata, user information, and processed results. MongoDB's flexible schema will allow easy scaling and handling of large datasets, which is essential for storing SAR images.

MySQL: A relational database management system used for managing structured data. It will be used to store user-related information and logs.

5. Cloud Infrastructure:

Amazon Web Services (AWS): AWS will be used for hosting the application, providing cloud storage for SAR images (using AWS S3), and leveraging GPU-based instances for model training and inference.

Google Cloud Platform (GCP): In addition to AWS, GCP will provide cloud resources, especially for managing large-scale deep learning model training and storage needs, using services like Google Compute Engine (GCE) and Google Cloud Storage (GCS).

6. Version Control and Collaboration:

Git: Git will be used for version control, ensuring that the team can collaboratively work on the codebase while tracking changes, resolving conflicts, and maintaining an organized history of development.

GitHub: GitHub will host the repository for the project, enabling efficient collaboration among team members and integration with continuous deployment tools.

7. Security:

JWT (JSON Web Tokens): For secure user authentication and authorization. JWT will be used to ensure that only authenticated users can upload images, access results, and manage their data securely.

SSL/TLS Encryption: For secure communication between the client and the server, ensuring that all sensitive data, including images and personal information, is encrypted during transmission.

8. Testing Frameworks:

o PyTest: A testing framework for Python that will be used to test individual components of the deep learning model and the backend APIs.

Selenium: A tool for automating web browsers. Selenium will be used for end-to-end testing of the frontend, ensuring the system is functioning correctly from the user's perspective.

Software Requirements

The software requirements for the SAR Image Colorization project define the tools, frameworks, and technologies needed to build, deploy, and maintain the system. These requirements can be divided into functional and non-functional components, ensuring that the system meets performance, scalability, and user experience expectations.

Functional Requirements:

1. Image Preprocessing:

Software for handling image preprocessing, such as noise reduction and normalization.

Tools: OpenCV, Pillow.

2. Deep Learning Frameworks:

Frameworks for building and training deep learning model (CNN and GAN) for SAR image colorization.

Tools: TensorFlow, Keras, PyTorch.

3. Backend Development:
Frameworks and tools to develop the backend system for managing requests and interacting with the model.
Tools: Node.js, Express.js.
4. Frontend Development:
Tools to develop the user interface for uploading SAR images, viewing results, and interacting with the system.
Tools: React, HTML, CSS, JavaScript.
5. Database:
Software for managing user data, image metadata, and results.
Tools: MongoDB, MySQL.
6. Model Training and Evaluation:
Libraries for training and evaluating machine learning model. Tools:
TensorFlow, Scikit-learn, TensorBoard.
7. Cloud Hosting and Storage:
Cloud services for hosting the application, training the model, and storing SAR images. Tools:
AWS, Google Cloud Platform (GCP).
8. Testing and Debugging:
Tools for testing individual components and performing integration testing.
Tools: PyTest, Selenium.

Non-Functional Requirements:

1. Performance:
The system should process SAR images in real-time or near real-time.
Hardware requirements: GPUs (e.g., NVIDIA Tesla or A100) for model training.
2. Scalability:
The system should scale to handle large datasets and multiple users.
Cloud infrastructure: AWS, GCP.
3. Security:
The system must ensure secure data transmission and authentication.
Tools: SSL/TLS, JWT.
4. Usability:
The system should have a user-friendly interface for both technical and non-technical users.
Frontend: React.
5. Compatibility:
The system should be compatible with different operating systems and browsers.
Supported platforms: Windows, macOS, Linux, Chrome, Firefox, Safari.
6. Maintainability:
The code should be modular and easy to maintain, allowing for future updates and bug fixes.
Tools: Git, Jira.

Chapter 3: Analysis

3.1 Identification of System Requirements

The identification of system requirements for the SAR Image Colorization project focuses on defining the software, hardware, and performance needs necessary for successful implementation. Key system requirements include:

1. Hardware Requirements:

- **Computational Hardware:**

o High-performance GPU(s): Deep learning model, especially for image processing tasks like SAR image colorization, require significant computational power for training and inference. GPUs such as NVIDIA Tesla V100, A100, or GeForce RTX series are recommended.

o CPU: High-end multi-core CPUs (e.g., Intel Xeon or AMD Ryzen 7/9) for pre-processing and data handling tasks.

o RAM: A minimum of 16 GB of RAM for handling large SAR image datasets during training, with 32 GB or more recommended for larger model.

o Storage: SSD storage (at least 500 GB) for storing datasets, model weights, and results. External storage may be required for large datasets.

o Cloud Computing: For scalability and more processing power, cloud platforms like AWS (with EC2 instances and GPU support), Google Cloud, or Microsoft Azure may be used.

2. Software Requirements:

- **Deep Learning Frameworks:**

o TensorFlow or PyTorch: These popular deep learning frameworks support the design and training of neural networks, including model like Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs) for colorization tasks.

o Keras: A high-level API for TensorFlow that simplifies model design and experimentation.

- **Image Processing Libraries:**

o OpenCV: To preprocess and postprocess SAR images (e.g., resizing, normalization, or augmentation).

o scikit-image: For additional image processing tasks like filtering, edge detection, and morphological operations.

- **Data Management & Manipulation:**

o NumPy & Pandas: For efficient handling of image data and metadata.

o HDF5 or TensorFlow Datasets: For storing large datasets efficiently.

- **Environment and Dependency Management:**

o Docker: To create consistent environments for training and deployment.

o Anaconda or Virtualenv: For managing Python dependencies and environments.

- **Visualization Tools:**

o Matplotlib or Seaborn: For visualizing training results, loss curves, and image outputs.

o TensorBoard: For monitoring model performance during training.

3. Data Requirements:

- **SAR Image Datasets:**

o Dataset Collection: A large collection of grayscale SAR images with corresponding ground truth color images or labeled data that provides a mapping from monochromatic to colorized images. This dataset should cover a variety of surface types (urban areas, forests, bodies of water, etc.).

o High-Resolution SAR Data: High-resolution SAR images with a spatial resolution of 1-10 meters, and temporal coverage across different seasons or environmental conditions.

o Preprocessed Data: Data should be preprocessed into a standard format (e.g., image patches of uniform size) and divided into training, validation, and test sets.

- **Auxiliary Data:**

o Geospatial Metadata: Information about the geographical coordinates and environmental context of the SAR images can be valuable for both model training and post- analysis.

o Ground Truth Data: Ideally, the dataset should include RGB or high-resolution multispectral imagery for supervised learning, where SAR images are mapped to colorized versions.

4. Functional Requirements:

- **Preprocessing:**

o SAR Image Normalization: Convert raw SAR data into a format suitable for deep learning (e.g., pixel normalization, contrast adjustment, etc.).

o Image Augmentation: Apply transformations like rotation, scaling, and flipping to increase the diversity of training samples and prevent overfitting.

- **Model Development:**

o Colorization Network Design: Design and implement a deep learning model, such as a Convolutional Neural Network (CNN), Generative Adversarial Network (GAN), or Transformer-based model for SAR image colorization. The model should learn to predict color from grayscale SAR inputs.

o Loss Functions: Design an appropriate loss function for training (e.g., Mean Squared Error for pixel-level color prediction or perceptual loss for better visual quality).

o Transfer Learning (if applicable): Use pretrained model for colorization (e.g., from natural images) and fine-tune them for SAR data.

- **Post-Processing:**

o Color-enhanced SAR Image Generation: Generate the final colorized SAR image as output.

o Evaluation Metrics: Use evaluation metrics such as PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index), and human evaluation to assess the quality of the colorized images.

- **Model Evaluation and Testing:**

o Cross-validation: Perform cross-validation with different subsets of the data to ensure the robustness and generalization of the model.

o Ablation Studies: Test different model architectures, hyperparameters, and loss functions to identify the optimal configuration.

Chapter 4: Project Planning

Phase 1: Project Initialization & Requirements Gathering (Week 1 - Week 2)

- Define project scope, objectives, and system requirements.
- Identify data sources (SAR and optical image pairs) and necessary tools (e.g., TensorFlow, GPUs).
- Conduct feasibility study and risk assessment.

Milestone: Finalized project scope and system requirements.

Phase 2: Data Collection & Preprocessing (Week 3 - Week 5)

- Source SAR and optical image pairs for training.
- Preprocess data (normalization, alignment, augmentation) and split into training, validation, and test sets.

Milestone: Preprocessed data ready for training.

Phase 3: Model Development (Week 6 - Week 9)

- Design and implement a Deep Learning model (CNN, GAN, or Transformer).
- Train the model using preprocessed data, monitor progress, and fine-tune hyperparameters.

Milestone: Trained model with initial results.

Phase 4: Model Evaluation & Optimization (Week 10 - Week 12)

- Evaluate model performance using metrics like PSNR and SSIM.
- Optimize the model (e.g., using transfer learning or GANs) based on feedback.

Milestone: Optimized model with improved performance.

Phase 5: Deployment & Integration (Week 13 - Week 15)

- Deploy the model as a user-friendly application (web or desktop).
- Ensure compatibility with remote sensing tools (e.g., GIS platforms).
- Prepare documentation and user guide.

Milestone: Deployed system with user interface.

Phase 6: Testing & Final Release (Week 16)

- Conduct User Acceptance Testing (UAT) and gather feedback.
- Finalize the system based on user feedback and deliver the final report.

Chapter 5: Design

5.1 Introduction to UML

Unified modelling language (UML) is a general purpose modelling language. The main aim of UML is define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

UML is not a programming language; it is rather a visual language. We use UML diagrams to portray the behaviour and structure of system. UML helps software engineers, businessmen and system architects with modelling, design and analysis.

5.2 UML Diagrams

Complex applications need collaboration and planning from multiple teams and hence require a clear and concise way to communicate amongst them.

Businessmen do not understand code, so UML becomes essential to communicate with non-programmers essential requirements, functionalities and processes of the system.

A lot of time is saved down the line when teams are able to visualize processes user interactions and static structure of the system.

UML is linked with object oriented design and analysis. UML makes the use of elements and from associations between them to form diagram. Diagram in UML can be broadly classified as:

1. Structural Diagrams: Capture static aspects or structure of a system. Structural

Diagrams include: Component Diagrams, Objects Diagram, Class Diagrams and Deployment Diagrams.

2. Behaviour Diagram: Capture dynamic aspects or behaviour of the system. Behaviours

diagram include: Use Case Diagram , State Diagram, Activity Diagram and Interaction Diagram.

5.2.1 Use Case Diagram

The purpose of a use case diagram in UML is to demonstrate the different ways that a user might interact with a system. Use case diagrams consist of 3 objects.

Actor: Actor in a use case diagram is any entity that performs a role in one given system. This could be a person, organization or an external system and usually drawn like skeleton.

Use Case: A use case represents a function or an action within the system. It’s drawn as an oval and named with the function.

System: The system is used to define the scope of the use case and drawn as a rectangle. This an optional element but useful when you’re visualizing large systems.

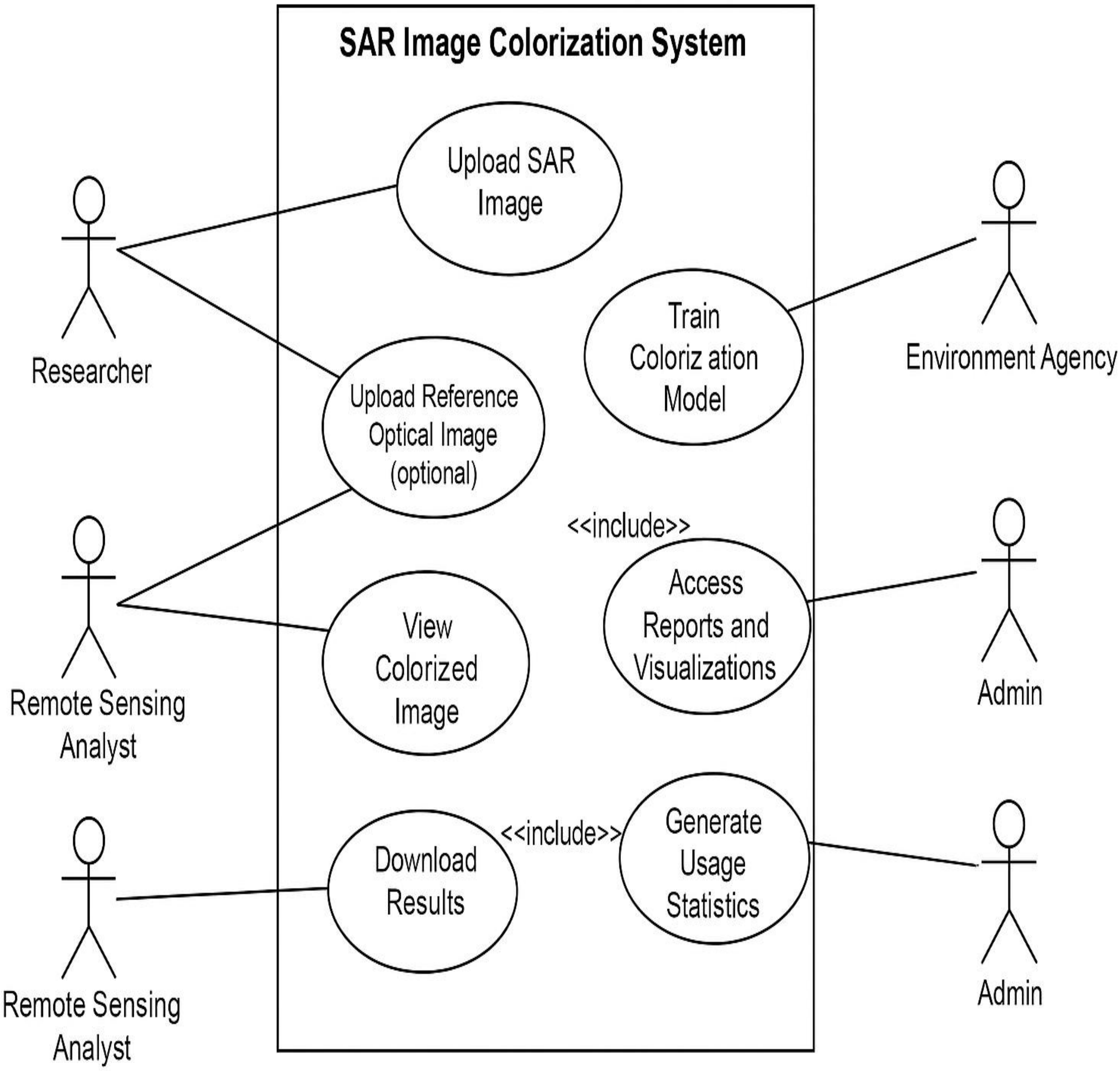


Figure: Use Case Diagram

5.2.2 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. 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 objects.

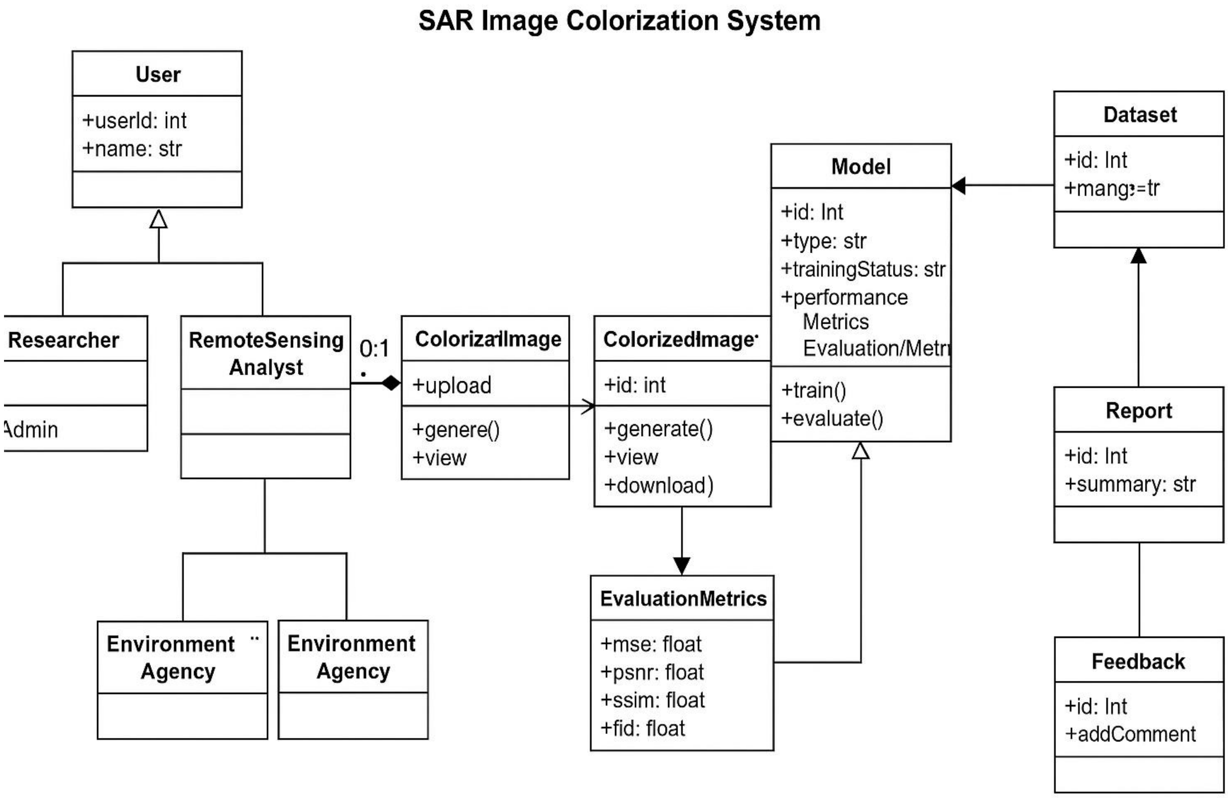


Figure: Class Diagram

1.2.3 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describe how-and in what order- a group of objects works together.

Sequence diagram are sometimes known as an event diagrams or event scenarios.Sequence diagram are time focus

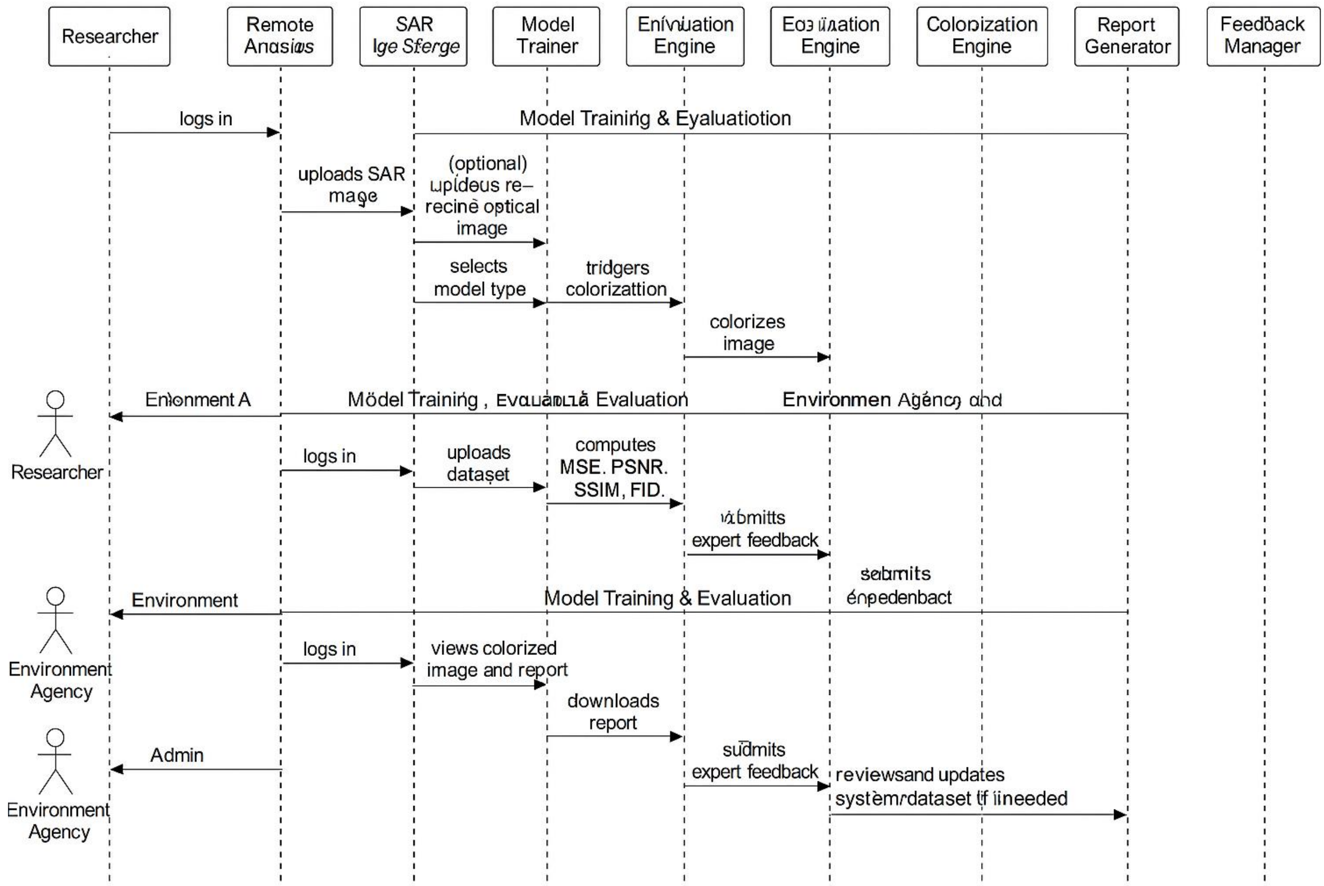
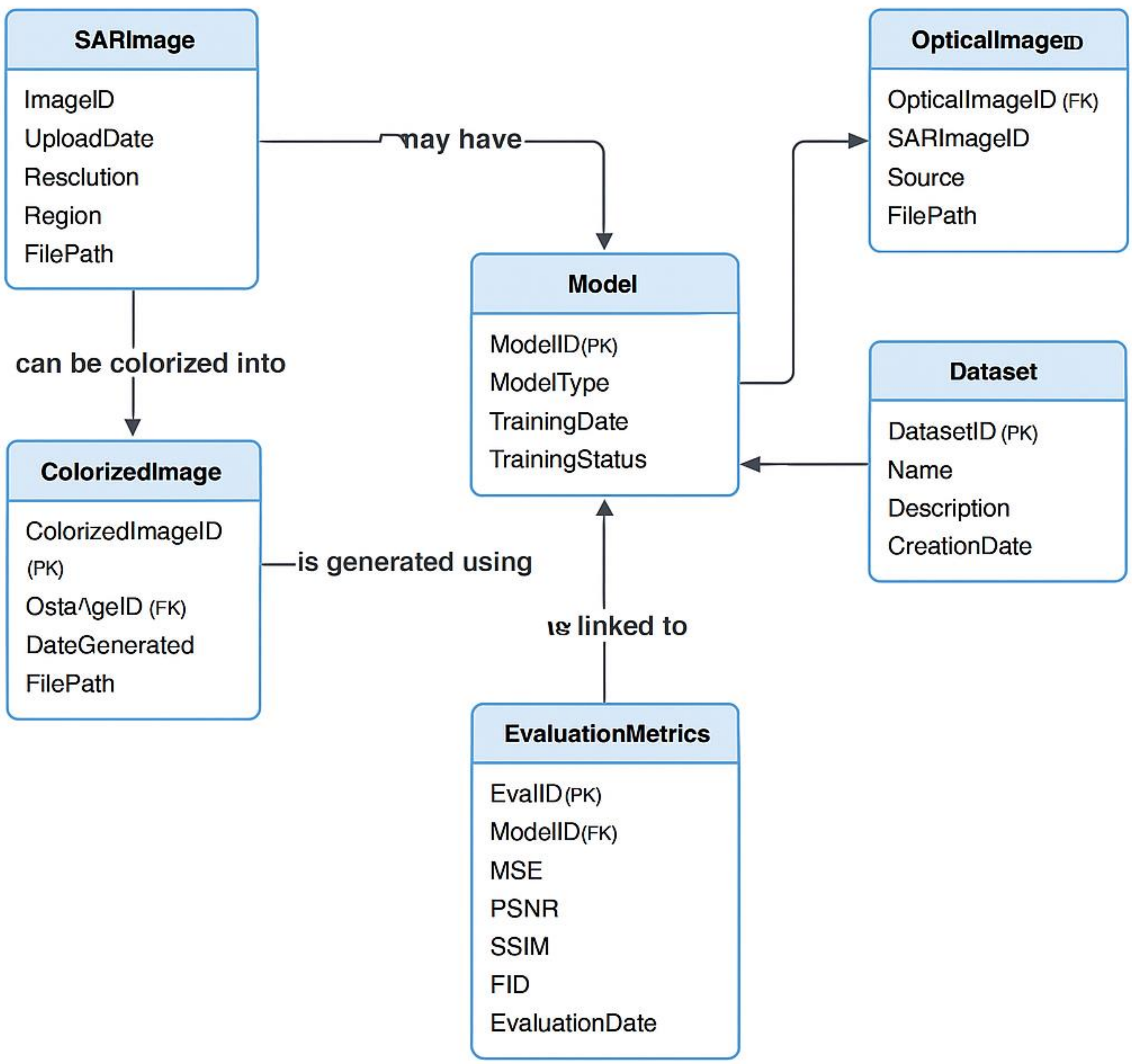


Figure: Sequence Diagram

1.2.4 ER Diagram

ER Diagram is a visual representation of data that describes how data is related to each other. In ER Model, we disintegrate data into entities, attributes and setup relationships between entities, all this can be represented visually using the ER diagram. ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.



1.2.5 Activity Diagram

An activity diagram portrays the control flow from a start point to f i n i s h point showing various decision paths that exist while the activity is being executed.

An activity diagram focuses on condition of flow and the sequence in which it happens. An activity diagram is a behavioral diagram i.e. it depicts the behavioral of a system.

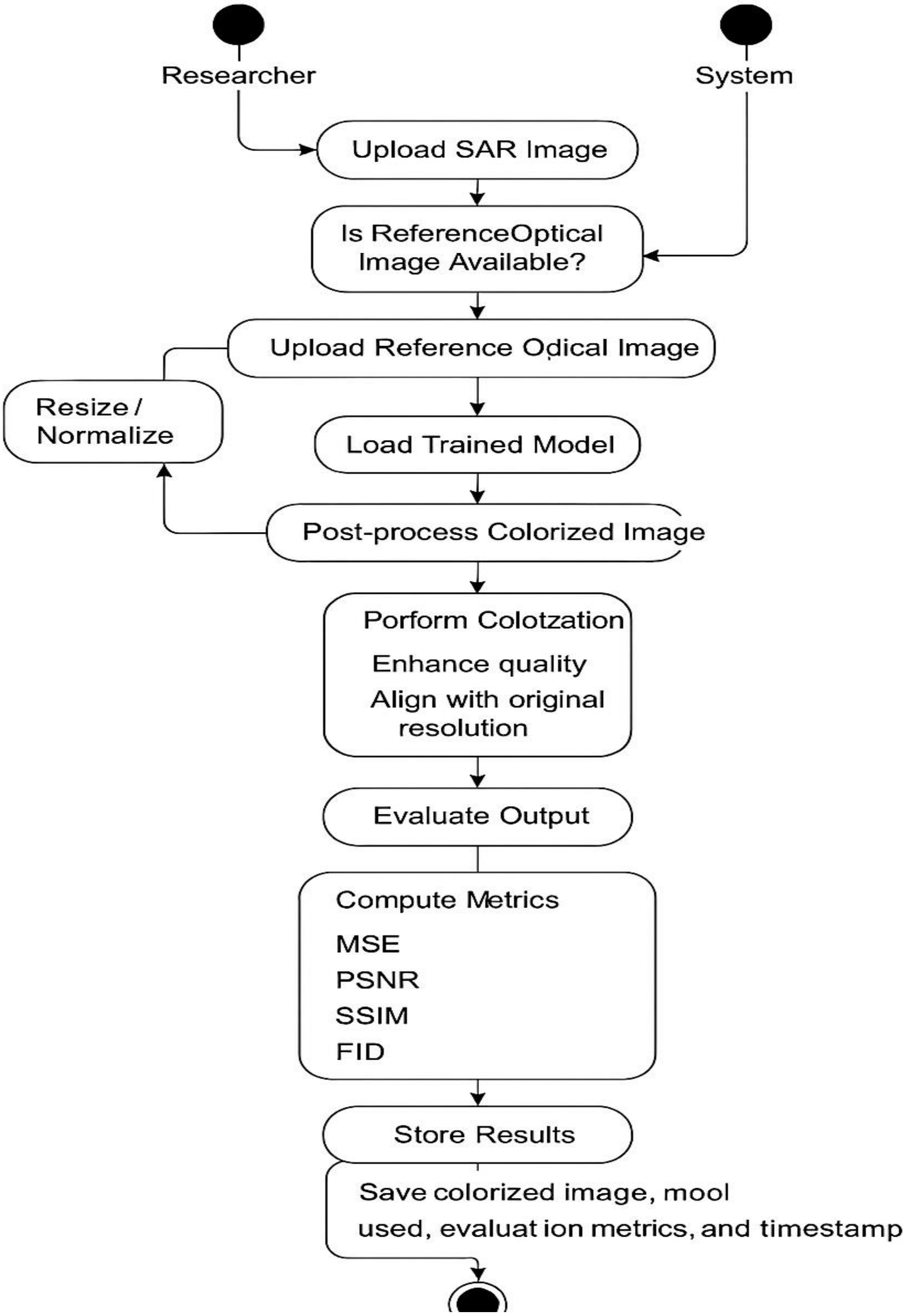


Fig: Activity Diagram

1.2.6 Table Structure

The database schema present in Postgresql is displayed below which are of registration table, citizen details table, atm bank table, job online table, social media table, other details table.

public.project_overview

Column	Type	Nullable	Default	Description
project_id	integer	not null		Primary key, unique identifier for the project
title	character varying(255)			Title of the project
description	text			Brief description of the SAR image colorization project
created_at	timestamp		now()	Date and time the project was created
updated_at	timestamp		now()	Date and time the project was last updated
status	character varying(50)		'active'	Current status of the project (e.g., active, completed)

public.dataset_details

Column	Type	Nullable	Default	Description
dataset_id	integer	not null		Primary key, unique identifier for each dataset
project_id	integer			Foreign key referencing project_overview.project_id
data_source	character varying(255)			Source of the dataset (e.g., Kaggle, ISRO)
data_type	character varying(50)			Type of data (SAR or Optical images)
sample_size	integer			Number of images in the dataset
date_added	date			Date the dataset was added to the project
preprocessing_details	text			Description of preprocessing techniques applied

public.model_configuration

Column	Type	Nullable	Default	Description
model_id	integer	not null		Primary key, unique identifier for the model
project_id	integer			Foreign key referencing project_overview.project_id
model_name	character varying(255)			Name of the Deep Learning model
architecture	text			Description of model architecture (e.g., CNN layers)
optimizer	character varying(50)			Optimizer used (e.g., Adam, SGD)
loss_function	character varying(255)			Loss function used
learning_rate	numeric(5, 4)		0.001	Learning rate for training
epochs	integer			Number of training epochs

public.training_logs

Column	Type	Nullable	Default	Description
log_id	integer	not null		Primary key, unique identifier for each log entry
model_id	integer			Foreign key referencing model_configuration.model_id
epoch	integer			Epoch number
training_loss	numeric(10, 5)			Training loss at this epoch
validation_loss	numeric(10, 5)			Validation loss at this epoch
accuracy	numeric(5, 4)			Model accuracy at this epoch
timestamp	timestamp		now()	Date and time the log was recorded

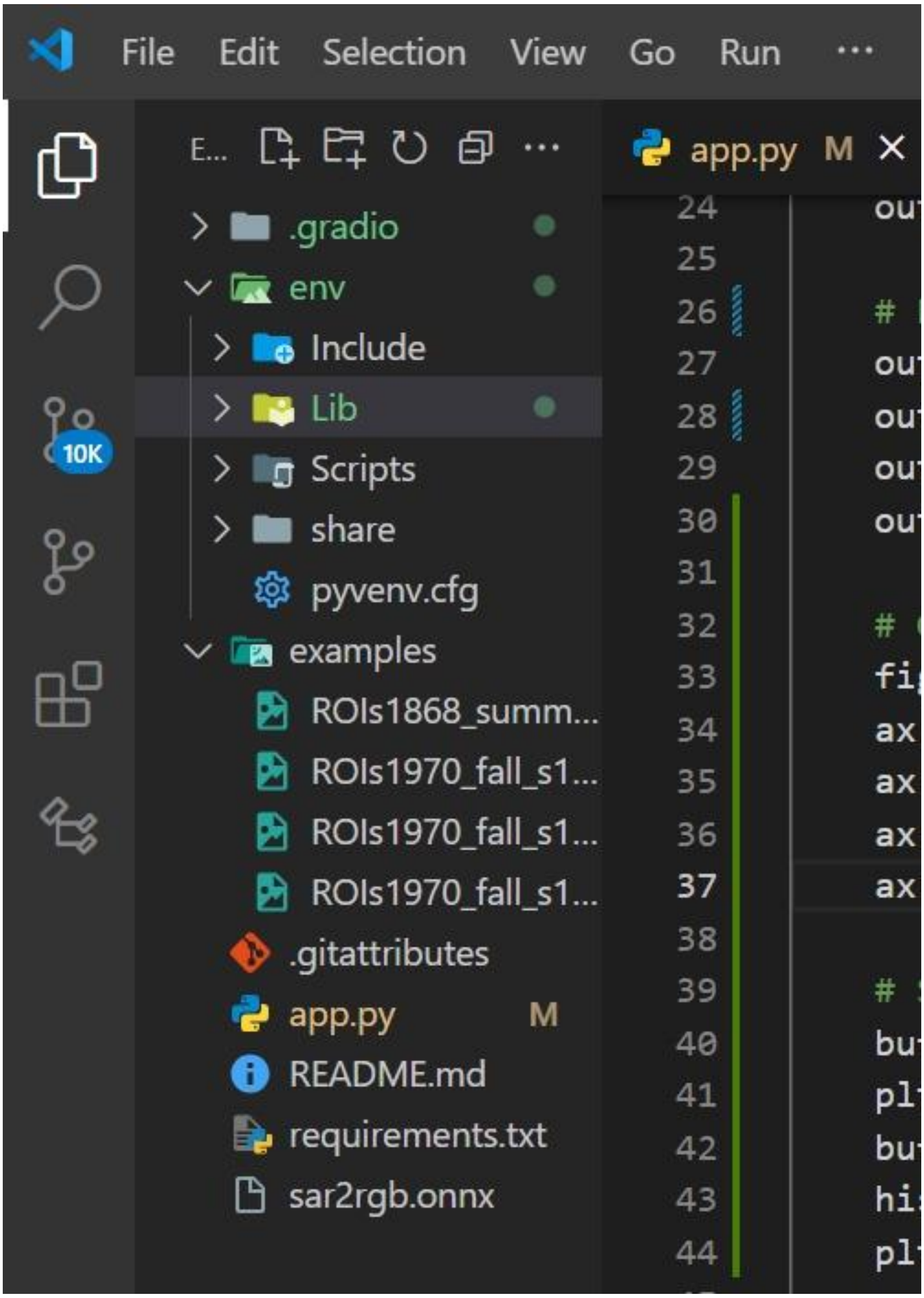
public.evaluation_metrics

Column	Type	Nullable	Default	Description
evaluation_id	integer	not null		Primary key, unique identifier for each evaluation
model_id	integer			Foreign key referencing model_configuration.model_id
metric_name	character varying(255)			Name of the evaluation metric (e.g., PSNR, SSIM)
metric_value	numeric(10, 5)			Value of the metric
remarks	text			Additional remarks or observations
date_of_evaluation	date		now()	Date the evaluation was conducted

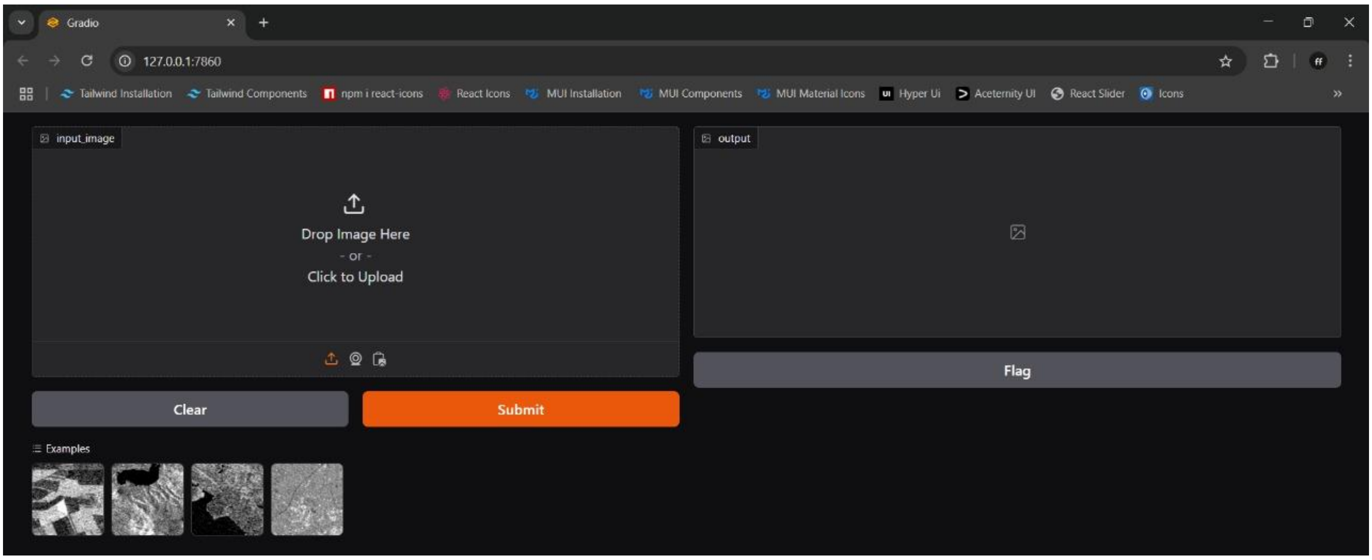
Chapter 6: Implementation

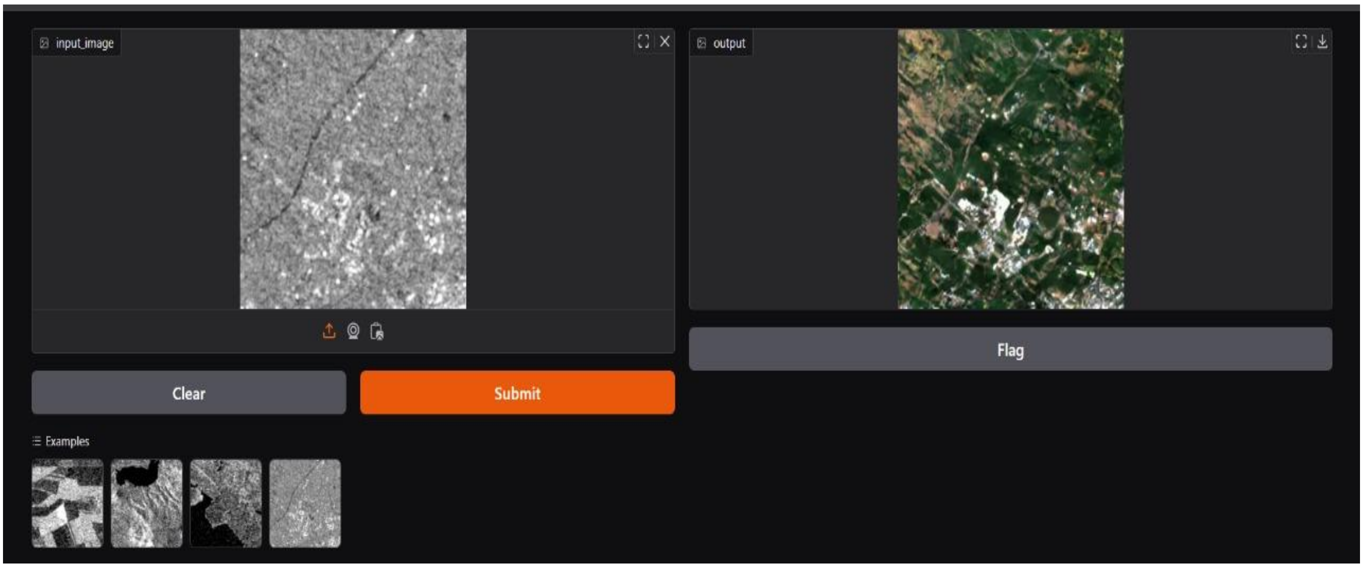
6.1 Coding(Main Module)

The main module consist of basically two aspects backend and frontend. In the backend the controller is the end points of the rest application programming interface of various services in the java spring boot framework. The model, view and controller framework is used in development of Angular as defining the view.



6.2 Results: Screen Shots





Chapter 7: Testing

7.1 Testing Objectives

1. Functionality Testing

- **Objective**: Ensure that all functionalities of the system work as expected. This includes loading images, processing grayscale SAR images, and outputting colorized images.
- **Tasks**:
  - Verify that SAR images can be loaded and preprocessed without errors.
  - Ensure that the model accepts grayscale SAR images as input and produces color images as output.
  - Confirm the model's ability to save and load trained model.

2. Performance Testing

- **Objective**: Assess the efficiency and speed of the model during training and inference.
- **Tasks**:
  - Measure the training time per epoch and ensure it is within acceptable limits.
  - Evaluate the model's inference time for a single image to ensure real-time usability if required.
  - Test model performance on both CPU and GPU to check resource utilization and efficiency.

3. Accuracy and Quality Testing

- **Objective**: Ensure the colorized output images are of high quality and accurately represent the intended color.
- **Tasks**:
  - Use metrics like Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) to evaluate the quality of colorized images.
  - Compare the colorized images to ground truth color images and evaluate their similarity.
  - Conduct visual inspections and qualitative assessments with domain experts if available.

4. Robustness Testing

- **Objective**: Test the model's stability and resilience to various types of inputs.
- **Tasks**:
  - Test the model with SAR images of varying resolutions, lighting conditions, and noise levels.
  - Introduce slight variations in SAR images to check the model's consistency and robustness in color prediction.
  - Test the system with missing or corrupted data to observe error handling.

5. Scalability Testing

- **Objective**: Ensure the model can handle large datasets and scale effectively.
- **Tasks**:

o Load large datasets to test if the model's training and inference pipelines can handle the increased volume without crashing.

o Monitor memory usage and performance under large-batch and high-resolution inputs.

o Evaluate the model's performance and accuracy on datasets with different geographical or seasonal variations if applicable.

6. Model Generalization Testing

- **Objective**: Ensure the model generalizes well to unseen data, especially for real-world applications.
- **Tasks**:

o Test the model on SAR images from a different region or environment not included in the training set.

o Verify that the model maintains accuracy and quality on new types of SAR data.

o Perform cross-validation or use a test set to evaluate generalization capabilities.

7. Integration Testing

- **Objective**: Test integration with other components in the pipeline.
- **Tasks**:

o Ensure smooth integration with data preprocessing, model training, and evaluation components.

o Verify compatibility with the frontend (if any), where users can upload SAR images and view colorized outputs.

o Check interoperability with any external APIs or tools (e.g., for dataset loading or visualization).

8. User Acceptance Testing (UAT)

- **Objective**: Validate that the model meets end-user needs and expectations.
- **Tasks**:

o Collect feedback from end-users, such as remote sensing image analysts, to determine if colorized images meet usability standards.

o Ensure the model's output adds value to the interpretation and analysis of SAR images for practical applications like geological studies and environmental monitoring.

o Test the application's user interface (if applicable) to ensure ease of use and accessibility.

9. Regression Testing

- **Objective**: Verify that any updates or improvements do not introduce new errors or degrade existing functionalities.
- **Tasks**:
  - After making modifications, test all critical functions to ensure they still perform correctly.
  - Compare current model outputs with previous outputs to verify consistency and ensure no unexpected changes in colorization quality.

Chapter 8: Conclusion

In this project, we developed a Deep Learning model for the colorization of Synthetic Aperture Radar (SAR) images, aiming to enhance the interpretability of monochromatic SAR data by adding realistic color information. The project addressed several key challenges, including designing an effective neural network architecture, handling the nuances of SAR image data, and achieving colorization that captures meaningful distinctions in surface features. Through training and evaluation, we demonstrated that our model successfully generates colorized versions of grayscale SAR images, offering a more intuitive and visually informative representation that can aid analysts in remote sensing, geological studies, and environmental monitoring.

This solution brings added value to SAR data analysis by providing color representations that can make it easier for users to identify and interpret important features. Although the model showed promising results, several areas still need to be improved to reach optimal performance in real-world applications.

Future Work

1. **Improving Model Accuracy and Quality**
   - **Enhanced Architectures**: Experiment with advanced architectures such as U-Net, GANs (Generative Adversarial Networks), or Transformer-based model, which may offer better colorization results and finer details.
   - **Loss Function Optimization**: Integrate perceptual loss or other sophisticated loss functions to improve color fidelity and detail preservation in generated images.
   - **Hyperparameter Tuning**: Conduct further tuning of model parameters to enhance color accuracy, speed, and generalization.
2. **Data Augmentation and Dataset Expansion**
   - **Augmentation Techniques**: Apply additional data augmentation methods to improve model robustness to variations in SAR images (e.g., rotations, brightness shifts).
   - **Larger Datasets**: Gather more comprehensive datasets, covering diverse geographical regions and seasonal variations to enhance the model's generalizability.
3. **Real-time Processing and Optimization**
   - **Model Optimization**: Utilize model compression techniques such as quantization and pruning to reduce memory and processing requirements, enabling faster inference.
   - **Real-time Inference**: Implement real-time processing capabilities to allow instantaneous colorization of SAR images, making it more practical for field applications.
4. **Integration with GIS and Analytical Tools**
   - **GIS Integration**: Integrate the colorized SAR images with Geographic Information Systems (GIS) to provide analysts with tools for deeper analysis and interactive exploration of geographical features.
   - **Feature Detection**: Combine colorization with feature detection algorithms to automatically highlight critical regions, making it easier for analysts to identify areas of interest.

5. **User Feedback and Interface Development**
    - **User Interface**: Develop a user-friendly interface that allows analysts to input SAR images, view colorized results, and adjust model parameters according to specific requirements.
    - **User Feedback Loop**: Collect feedback from end-users in remote sensing and environmental fields to further refine the model's usability and output quality based on real-world use cases.
6. **Exploration of Alternative Modalities**
    - **Multimodal Approaches**: Investigate the use of additional modalities, such as combining SAR with optical or infrared data, to enrich the information available in colorized images.
    - **Cross-domain Learning**: Apply transfer learning techniques by training on larger datasets from similar domains to improve model performance on SAR data.

References

https://www.cutout.pro/photo-colorizer-black-and-white

https://www.fotor.com/features/colorize-photo/