

**Software Requirements Specification**

# **Change Detection Method for Remote Sensing Images**

**GROUP 6**

**SHUBHENDRA GAUTAM IIT2021142**

**SARANSH YADAV IIT2021162**

**YASH SINGH IIT2021165**

**VISHAL KUMAR IIT2021196**

**VIVEK MANWAR IIT2021206**

**SAIKAT SADHUKHAN IIT2021261**

**1. Introduction**

**1.1 Purpose:**

The purpose of this document is to provide a comprehensive overview of the requirements and specifications for the development of a Change Detection Method for Remote Sensing Images Based on Coupled Dictionary and Deep Learning. It aims to define the features, functionalities, constraints, and interfaces of the proposed system to guide its development process effectively.

**1.2 Document Conventions:**

* Requirement IDs: Each requirement is assigned a unique identifier following the format "R1", "R2", etc., for easy reference and tracking.
* Functional Requirements: Functional requirements are expressed using "shall" statements to denote system capabilities and behaviors.
* Nonfunctional Requirements: Nonfunctional requirements are categorized separately to address aspects such as performance, safety, security, and software quality.
* Use Cases: Use cases are presented in a narrative format to describe system interactions and behaviors from the perspective of different stakeholders.

**1.3 Intended Audience and Reading Suggestions:**

This document is intended for various stakeholders involved in the development and implementation of the Change Detection Method, including developers, testers, project managers, and end users. It is assumed that readers have a basic understanding of remote sensing, image processing, and machine learning concepts to comprehend the technical details presented in this document.

**1.4 Project Scope:**

The scope of this project encompasses the design and development of a software system capable of accurately detecting changes in remote sensing images using a combination of coupled dictionary learning and deep learning techniques. The system will analyze pairs of input images captured at different time intervals and generate a binary map indicating change/no change regions within the scene.

**1.5 References:**

* J. G. Vinholi, B. G. Palm, D. Silva, R. Machado and M. I. Pettersson, "Change Detection Based on Convolutional Neural Networks Using Stacks of Wavelength-Resolution Synthetic Aperture Radar Images," in IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1-14, 2022, Art no. 5236414, doi: 10.1109/TGRS.2022.3211010.
* Yang, W., Song, H., Du, L., Dai, S., & Xu, Y. (2022, January 17). A Change Detection Method for Remote Sensing Images Based on Coupled Dictionary and Deep Learning. Computational Intelligence and Neuroscience, 2022, 1–14. https://doi.org/10.1155/2022/3404858
* Shafique, A., Cao, G., Khan, Z., Asad, M., & Aslam, M. (2022, February 11). Deep Learning-Based Change Detection in Remote Sensing Images: A Review. Remote Sensing, 14(4), 871. https://doi.org/10.3390/rs14040871
* JCheng, Guangliang & Huang, Yunmeng & Li, Xiangtai & Lyu, Shuchang & Xu, Zhaoyang & Zhao, Qi & Xiang, Shiming. (2023). Change Detection Methods for Remote Sensing in the Last Decade: A Comprehensive Review.

**2. Overall Description**

**2.1 Product Perspective:**

The Change Detection Method for Remote Sensing Images will be developed as a standalone software system that integrates seamlessly with existing remote sensing platforms and tools. It will serve as a complementary tool for analysts and researchers to perform change detection tasks efficiently and accurately.

**2.2 Product Features:**

The key features of the system include:

* Input image preprocessing to enhance quality and reduce noise.
* Feature extraction using coupled dictionary learning to capture discriminative patterns.
* Deep learning based change detection for accurate identification of changes.
* Postprocessing techniques for refining change detection results and reducing false alarms.

**2.3 User Classes and Characteristics:**

The system is designed to cater to the following user classes:

* Remote sensing researchers: Users with expertise in remote sensing and image analysis.
* GIS professionals: Users involved in geographic information system (GIS) applications.
* Environmental scientists: Users interested in monitoring environmental changes.
* Disaster management authorities: Users responsible for assessing disaster impacts.
* Urban planners: Users engaged in urban development and monitoring activities.

**2.4 Operating Environment:**

The system is platformindependent and can be deployed on various operating systems, including Windows, Linux, and macOS. It is implemented using the Python programming language and relies on libraries such as OpenCV, TensorFlow, and scikitlearn for image processing and machine learning tasks.

**2.5 Design and Implementation Constraints:**

The system must adhere to the following constraints:

* Computational resources: Sufficient computational resources are required for deep learning model training and inference.
* Labeled training data: Availability of labeled training data is necessary to train accurate deep learning models.
* Compatibility with data formats: The system should support common remote sensing data formats to ensure interoperability with existing datasets and tools.

**2.6 Assumptions and Dependencies:**

* The successful development and operation of the system are based on the following assumptions and dependencies:
* Availability of remote sensing image datasets for training and evaluation purposes.
* Access to suitable computing infrastructure for model training, validation, and deployment.
* Compliance with data privacy and security regulations governing the handling of sensitive information.

**3. System Features**

**3.1 Functional Requirements:**

* **Feature Extraction (R1):**

The system shall extract relevant features from input remote sensing images using coupled dictionary learning techniques to capture the underlying structure of the scene effectively.

* **Change Detection (R2):**

The system shall employ deep learning algorithms to detect changes between pairs of input images captured at different time intervals, producing a binary map indicating change/nochange regions.

* **Postprocessing (R3):**

The system shall apply postprocessing techniques, such as morphological operations and spatial filtering, to refine the change detection results and reduce false alarms caused by noise or artifacts.

**4. External Interface Requirements**

**4.1 User Interfaces:**

The system shall provide the following user interfaces:

* Graphical User Interface (GUI): A userfriendly interface for interacting with the system, including options for inputting images, configuring parameters, and visualizing results.
* Commandline Interface (CLI): A commandline interface for batch processing and automation, allowing users to execute the system's functionality in a scriptable manner.

**4.2 Hardware Interfaces:**

The system shall be compatible with standard hardware configurations commonly used for image processing and machine learning tasks, including CPU, GPU, and memory specifications.

**4.3 Software Interfaces:**

The system shall integrate with existing remote sensing software platforms through Application Programming Interfaces (APIs) or support common data formats for seamless data exchange and interoperability.

**4.4 Communications Interfaces:**

The system shall support data exchange with remote sensing data repositories and web services for accessing additional image datasets or external resources required for analysis.

**5. Nonfunctional Requirements**

**5.1 Performance Requirements:**

* The system shall process input images within a reasonable time frame, with typical satellite images processed in minutes.
* The change detection accuracy shall meet or exceed the performance of existing methods, with a high detection rate and low false alarm rate.

**5.2 Safety Requirements:**

* The system shall ensure the integrity and confidentiality of input image data, implementing appropriate security measures to prevent unauthorized access or data breaches.
* Error handling mechanisms shall be implemented to detect and recover from system errors or failures, minimizing disruptions and ensuring data integrity.

**5.3 Security Requirements:**

* The system shall implement authentication and authorization mechanisms to control access to sensitive data and features, ensuring that only authorized users can perform certain actions or access specific resources.
* Data encryption shall be employed to protect confidential information stored or transmitted by the system, preventing unauthorized interception or tampering.

**5.4 Software Quality Attributes:**

* **Maintainability**: The system shall be designed with modularity and extensibility to facilitate future updates, maintenance, and enhancements, allowing for the addition of new features or improvements.
* **Usability**: The user interface shall be intuitive and user friendly, providing clear instructions, feedback, and error messages to guide users through the system's functionality effectively.
* **Reliability**: The system shall demonstrate robustness to variations in input data and operating conditions, minimizing errors, false detections, and inaccuracies in change detection results.