

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

# **Group 6 : Design Report**

**SHUBHENDRA GAUTAM IIT2021142**

**SARANSH YADAV IIT2021162**

**YASH SINGH IIT2021165**

**VISHAL KUMAR IIT2021196**

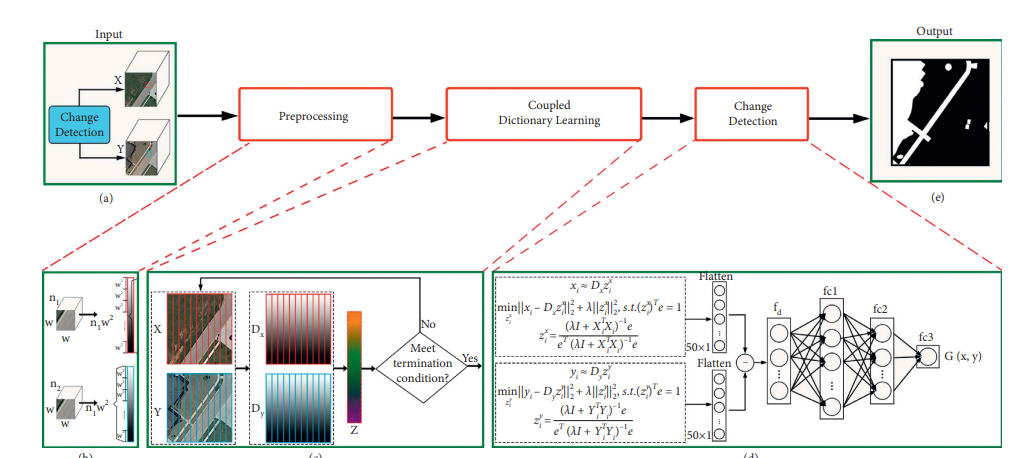
**VIVEK MANWAR IIT2021206**

**SAIKAT SADHUKHAN IIT2021261**

## **1. Introduction**

The introduction section sets the context for the design report, outlining the purpose, document conventions, intended audience, project scope, and references. It serves as a primer for stakeholders to understand the significance and scope of the system being designed.

## **2. System Architecture**



### 

### **2.1 Input Image Preprocessing Module**

#### Techniques:

* Image Denoising: Utilizes algorithms such as Gaussian blur or median filtering to remove noise from input images.
* Contrast Enhancement: Adjusts image contrast using techniques like histogram equalization to improve visual quality and feature representation.

### **2.2 Feature Extraction Module**

#### Techniques:

* Coupled Dictionary Learning: Employs algorithms to learn a joint representation of paired input images, capturing shared and distinct features between them.
* Feature Vectorization: Converts learned features into compact, discriminative vectors suitable for input to the change detection module.

### **2.3 Deep Learning-Based Change Detection Module**

#### Architecture:

* Convolutional Neural Networks (CNNs): Adopts CNN architectures tailored for change detection tasks, possibly utilizing pretrained models for feature extraction.
* Training Strategy: Trains the model on pairs of input images using techniques like siamese networks or temporal differencing to learn temporal changes effectively.

### **2.4 Postprocessing Module**

#### Techniques:

* Morphological Operations: Performs dilation and erosion to remove isolated noise and smooth the change detection map.
* Spatial Filtering: Applies techniques like median filtering or Gaussian smoothing to further refine the results and reduce false alarms.

## **3. User Interfaces**

#### **3.1 Graphical User Interface (GUI)**

The graphical user interface (GUI) provides a user-friendly and interactive platform for users to interact with the system. It offers the following features:

##### **Image Selection:**

Users can easily load input images for change detection analysis by navigating through their local file system or network directories. The GUI presents a file explorer interface, allowing users to select one or multiple image pairs conveniently. Supported image formats include commonly used remote sensing formats such as GeoTIFF or JPEG.

##### **Parameter Configuration:**

To accommodate different scenarios and user preferences, the GUI allows users to adjust various preprocessing and postprocessing parameters. These parameters include but are not limited to:

* Image denoising parameters (e.g., filter types, kernel sizes)
* Contrast enhancement settings (e.g., histogram equalization methods, contrast stretching thresholds)
* Postprocessing options (e.g., morphological operation parameters, spatial filtering kernel sizes)

Users can modify these parameters through intuitive controls such as sliders, dropdown menus, or input fields. The GUI dynamically updates the parameter settings, providing immediate feedback to users.

##### **Result Visualization:**

Once the change detection analysis is complete, the GUI displays the results interactively to users. This includes:

* Change/no-change maps: Visual representations of regions where changes have been detected compared to the input images.
* Visualizations of detected regions: Highlighted areas or overlays on the input images to indicate the location and extent of detected changes.

Users can navigate through the visualization interface, zoom in/out, and pan across the images to examine the results in detail. Additionally, the GUI may provide options to export the visualized results in various formats for further analysis or reporting.

#### 3.2 Command-Line Interface (CLI)

The command-line interface (CLI) offers a more scriptable and automation-friendly way to interact with the system. It provides the following functionality:

##### **Batch Processing:**

The CLI facilitates the automated analysis of multiple image pairs in batch mode. Users can specify directories containing sets of input image pairs, and the CLI iterates through each pair, performing change detection analysis sequentially or in parallel, depending on system capabilities and user preferences.

Batch processing allows users to process large datasets efficiently without manual intervention, making it suitable for tasks such as monitoring changes over extensive geographical areas or processing time series of remote sensing imagery.

##### **Scriptable Execution:**

The CLI supports scriptable execution, enabling users to integrate the system into custom scripts or workflows. By providing command-line arguments or configuration files, users can specify parameters, input/output directories, and other settings required for change detection analysis.

Scriptable execution enhances the system's interoperability with existing automation frameworks, GIS pipelines, or remote sensing processing chains. Users can incorporate the CLI commands into scheduled tasks, automated workflows, or deployment scripts, streamlining the integration of change detection capabilities into broader data processing pipelines.

## **4. Hardware and Software Interfaces**

### **4.1 Hardware Interfaces**

#### Requirements:

* CPU and GPU Compatibility: Utilizes CPU and GPU resources for efficient image processing and deep learning computations.
* Memory Allocation: Ensures sufficient memory resources for handling large-scale remote sensing datasets and model parameters.

### **4.2 Software Interfaces**

#### Integration:

* API Integration: Integrates with existing remote sensing software platforms via APIs to enable seamless data exchange and interoperability.
* Data Format Support: Supports common remote sensing data formats such as GeoTIFF or ENVI to facilitate integration with diverse datasets and tools.

## **5. Nonfunctional Requirements**

### **5.1 Performance Requirements**

#### Metrics:

* Processing Time: Targets processing times within minutes for typical satellite imagery datasets.
* Accuracy Metrics: Aims for high detection rates and low false alarm rates to ensure reliable change detection performance.

### **5.2 Safety and Security Requirements**

#### Measures:

* Data Integrity: Implements measures to ensure the integrity and confidentiality of input data, including access controls and encryption.
* Error Handling: Incorporates robust error handling mechanisms to mitigate risks and ensure system stability.

### **5.3 Software Quality Attributes**

#### Emphasis:

* Maintainability: Designs the system with modular and extensible components to facilitate future updates and enhancements.
* Usability: Prioritizes user experience with intuitive interfaces and clear feedback mechanisms.
* Reliability: Strives for robust performance under varying conditions to minimize errors and inaccuracies in change detection results.

## **6. Conclusion**

The design report provides a detailed overview of the system architecture, user interfaces, hardware/software interfaces, and nonfunctional requirements for the Change Detection Method for Remote Sensing Images. By adhering to these design considerations, the system aims to provide accurate and reliable change detection capabilities for various remote sensing applications.