Detection and Localization of Tampering in Medical Images

Analysis Model

Submitted by:

Muskan Chalana	102203274
Ishita Jindal	102203668
Asmi Gaurav	102203253
Sudikshya Nyachhyon	102217201
Bhavuk Gupta	102203981

BE Third Year- COE & CSE

CPG No. 125

Under the Mentorship of

Dr. Shalini Batra

(Professor)

Dr. Geeta Kasana

(Assistant Professor)

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology, Patiala
February 2025

TABLE OF CONTENTS

1. Product Perspective ...…………………………...……………………………3
2. Use Case Diagrams...…………………………...……………………………4
3. Task Identification...………………………...……………………………10
4. Process Flow...…………………………...………………………………..12
5. Work Breakdown...………………………...……………………………...14
6. Scheduling...…………………………...……………………………………15
7. Requirements...…………………………...…………………………………16

1. Product Perspective

The Medical Image Tampering Detection System is designed to analyse and verify the authenticity of medical scans, ensuring protection against digital manipulation. The system is intended to serve as a forensic tool for radiologists, forensic analysts, and researchers, while also being accessible to patients and the general public for independent verification of scans. By detecting and localizing tampered regions in medical images, the system aims to prevent misdiagnoses, manipulated research findings, and unethical medical practices.

The core components of the system include:

1. **Input Module**: Receives medical images from various modalities (MRI, CT, mammograms etc.).
2. **Preprocessing Module**: Standardizes images, enhances quality, and extracts critical Regions of Interest (ROIs).
3. **Tampering Detection Module**: Analyses the image using multiple forensic and AI-based techniques to detect inconsistencies.
4. **Localization & Visualization Module**: Highlights tampered regions and generates interpretability maps for better analysis.
5. **Evaluation & Reporting Module**: Provides confidence scores, metrics, and a structured report for end-users.

This system is designed to be adaptable across different medical imaging modalities (MRI, CT, mammograms, etc.), making it a versatile and essential tool for forensic analysis, medical research, and public verification.

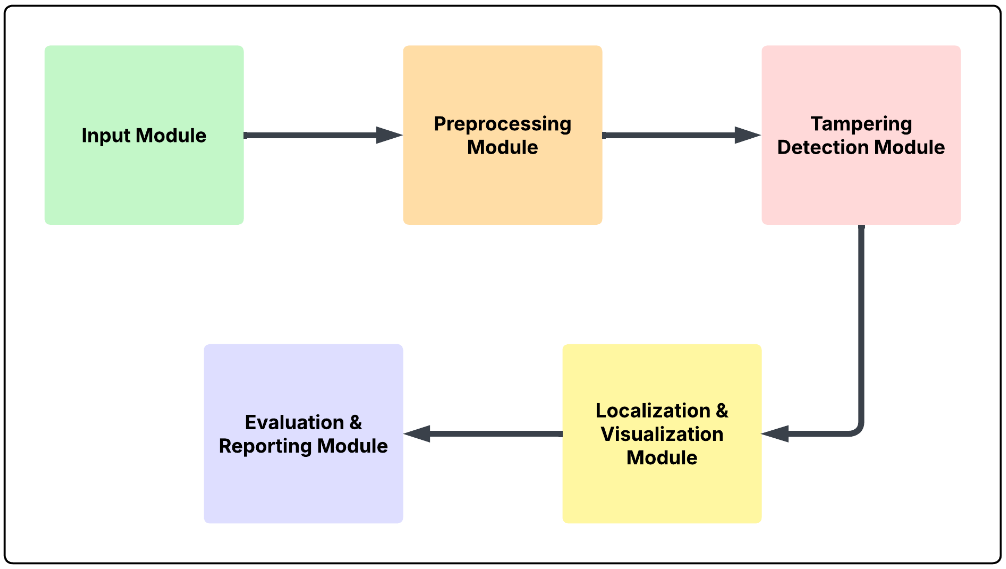

Input Module
Preprocessing Module
Tampering Detection Module
Localization & Visualization Module
Evaluation & Reporting Module

Fig 1. Block Diagram

2. Use case diagram

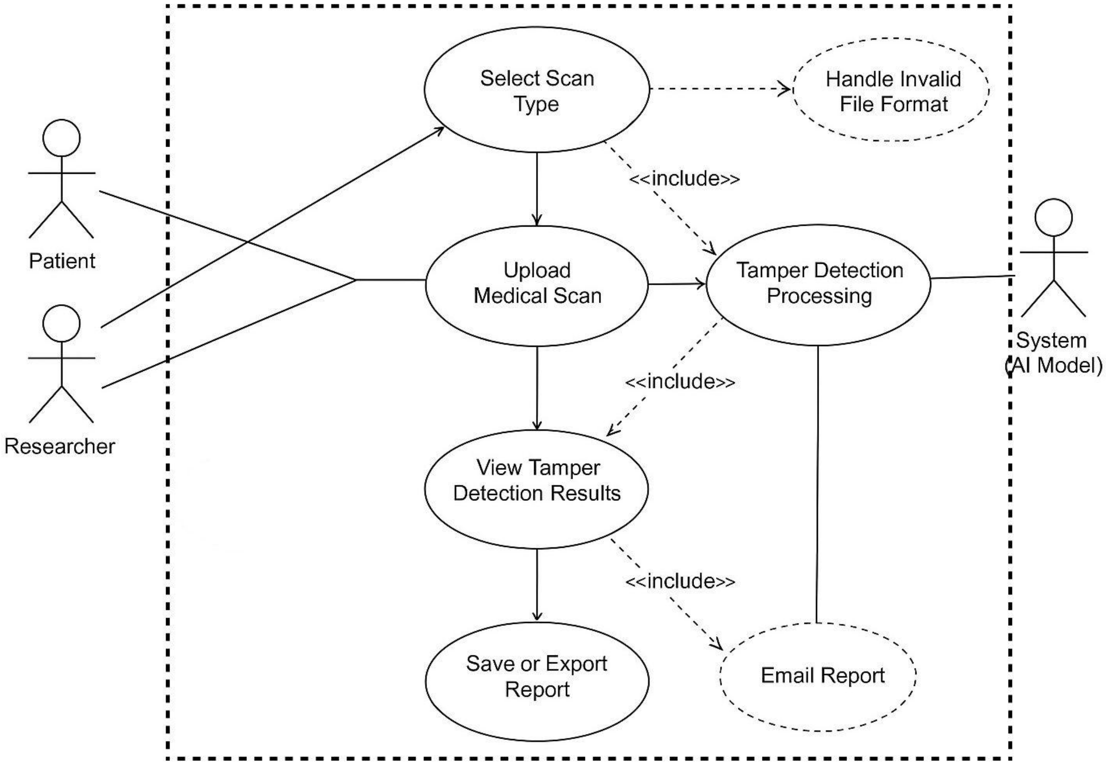


Fig 2. Use Case Diagram

Use Case 1: Upload Medical Scan

Use Case Name	Upload Medical Scan
Actors	Patient, Researcher
Pre-Conditions	User must be authenticated. The system should be ready to accept file uploads.
Normal Scenario	1. User selects the scan type.2. User uploads the medical scan.3. The system validates the file format.4. If valid, the scan is processed for tamper detection.5. The system stores the uploaded scan.
Alternate Flow	If the file format is invalid, the system triggers the "Handle Invalid File Format" use case.
Extension Points	Tamper Detection Processing
Post-Conditions	The scan is uploaded successfully and is ready for tamper detection processing.

Use Case 2: Tamper Detection Processing

Use Case Name	Tamper Detection Processing
Actors	System (AI Model)
Pre-Conditions	A valid medical scan must be uploaded.
Normal Scenario	1. The system receives the uploaded scan.2. The AI model processes the scan for tampering.3. The system generates detection results.
Alternate Flow	If the processing fails, an error is logged, and the user is notified.
Extension Points	View Tamper Detection Results
Post-Conditions	The tamper detection results are generated and stored.

Use Case 3: View Tamper Detection Results

Use Case Name	View Tamper Detection Results
Actors	Patient, Researcher
Pre-Conditions	The tamper detection process must be completed.
Normal Scenario	1.User requests to view the tamper detection results.2. The system retrieves the results.3. The results are displayed to the user.
Alternate Flow	If results are unavailable, an error message is shown.
Extension Points	Save or Export Report
Post-Conditions	The user has viewed the tamper detection results.

Use Case 4: Save or Export Report

Use Case Name	Save or Export Report
Actors	Patient, Researcher
Pre-Conditions	Tamper detection results must be available.
Normal Scenario	1.User chooses to save or export the report.2. The system generates a downloadable file.3. The file is saved locally or shared via email.
Alternate Flow	If saving/exporting fails, an error message is shown.
Extension Points	Email Report
Post-Conditions	The report is successfully saved or exported.

Use Case 5: Handle Invalid File Format

Use Case Name	**Handle Invalid File Format**
Actors	System
Pre-Conditions	A user must have attempted to upload an invalid file format.
Normal Scenario	1. The system detects an invalid file format.2. The system notifies the user of the error.3. The user is prompted to upload a valid file.
Alternate Flow	None
Extension Points	None
Post-Conditions	The user is informed of the issue and can retry the upload.

3. Complete Tasks and Sub-Tasks

Phase 1: Problem Definition & Research

- **1.1 Understanding Image Tampering in Medical Imaging**
  - Research common tampering techniques (splicing, blurring, noise addition, GAN-based modifications).
  - Study forensic methods used in medical image authentication.
- **1.2 Defining the Scope**
  - Select medical image types (lungs, rib cage, mammography).
  - Identify datasets (e.g., NIH Chest X-ray, DDSM for mammography).
  - Define project objectives and key performance metrics (accuracy, recall, AUC-ROC).

Phase 2: Data Collection & Preprocessing

- **2.1 Dataset Collection**
  - Acquire publicly available datasets.
  - Simulate tampered images using Photoshop, GANs, or custom modifications.
- **2.2 Data Preprocessing**
  - Normalize images (resize, grayscale conversion, histogram equalization).
  - Remove artifacts (noise reduction, contrast enhancement).
  - Data augmentation (rotation, flipping, zoom, adding noise).
- **2.3 Data Labeling**
  - Manually annotate tampered vs. non-tampered images.
  - Validate dataset with medical experts (if possible).

Phase 3: Model Development

- **3.1 Choosing Baseline Approaches**
  - Use conventional forensic techniques (Error Level Analysis, Noise Inconsistency, Frequency Domain Analysis).
- **3.2 Implement Deep Learning Models**
  - Experiment with CNNs, Autoencoders, or Vision Transformers.
  - Try pre-trained models like ResNet, EfficientNet, or UNet.
  - Apply attention mechanisms (e.g., ViT, Transformers).
- **3.3 Feature Engineering & Explainability**
  - Extract tampering features using Grad-CAM or SHAP.
  - Generate activation maps for interpretability.

Phase 4: Model Training & Evaluation

- **4.1 Train Models on Labeled Data**
  - Split into training, validation, and test sets.

    - Use hyperparameter tuning techniques.
- **4.2 Evaluate Model Performance**
    - Use metrics (accuracy, precision, recall, F1-score, AUC-ROC).
    - Perform cross-validation.
- **4.3 Compare Against Other Methods**
    - Compare performance with traditional forensic and AI-based methods.
    - Conduct ablation studies.

Phase 5: Deployment & Testing

- **5.1 Develop API or GUI**
    - Build a Flask/Django-based API or a Streamlined Web App.
- **5.2 Real-world Testing**
    - Test with unseen tampered images.
    - Collect feedback from domain experts.
- **5.3 Documentation & Report**
    - Prepare final documentation.

4. Process Flow

Swimlane Diagram

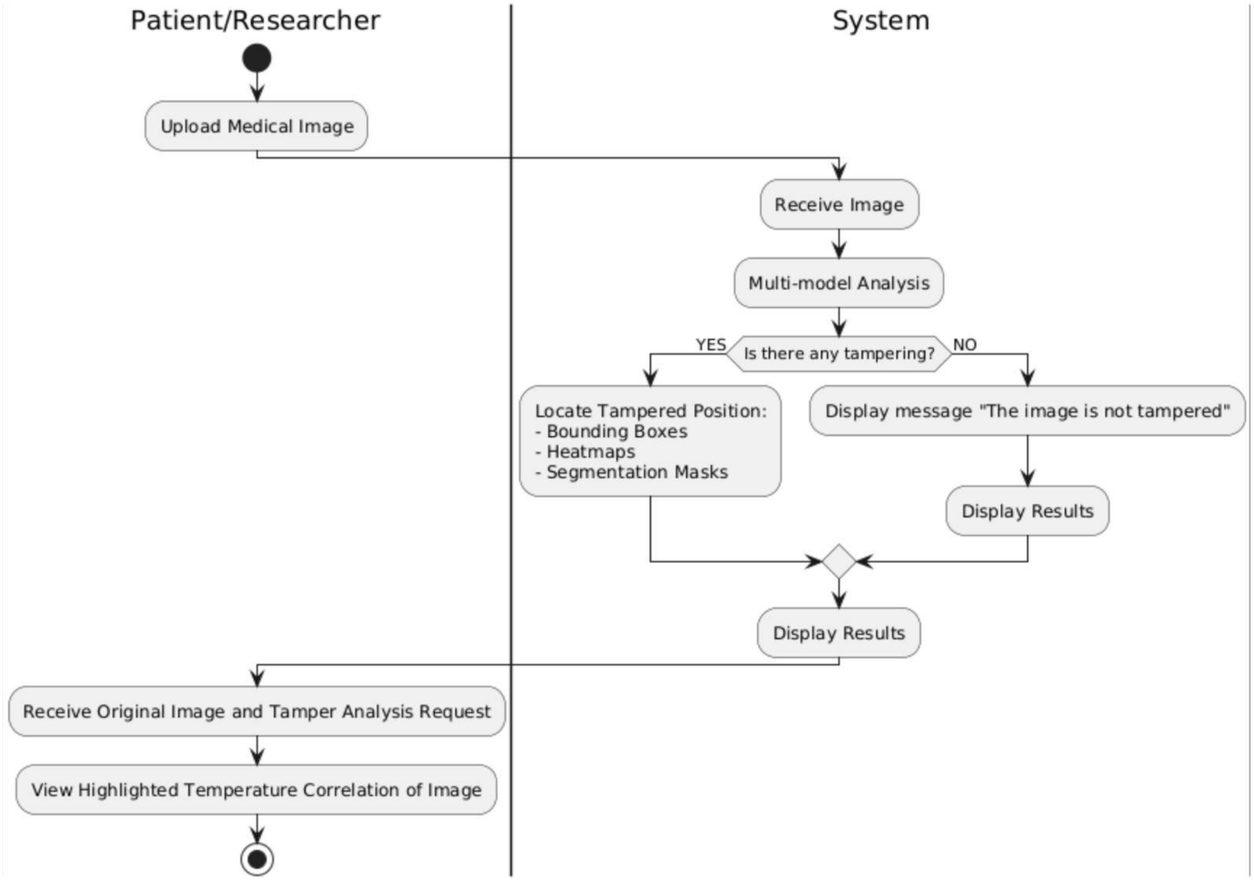


Fig 3. Swimlane Diagram

Activity Diagram

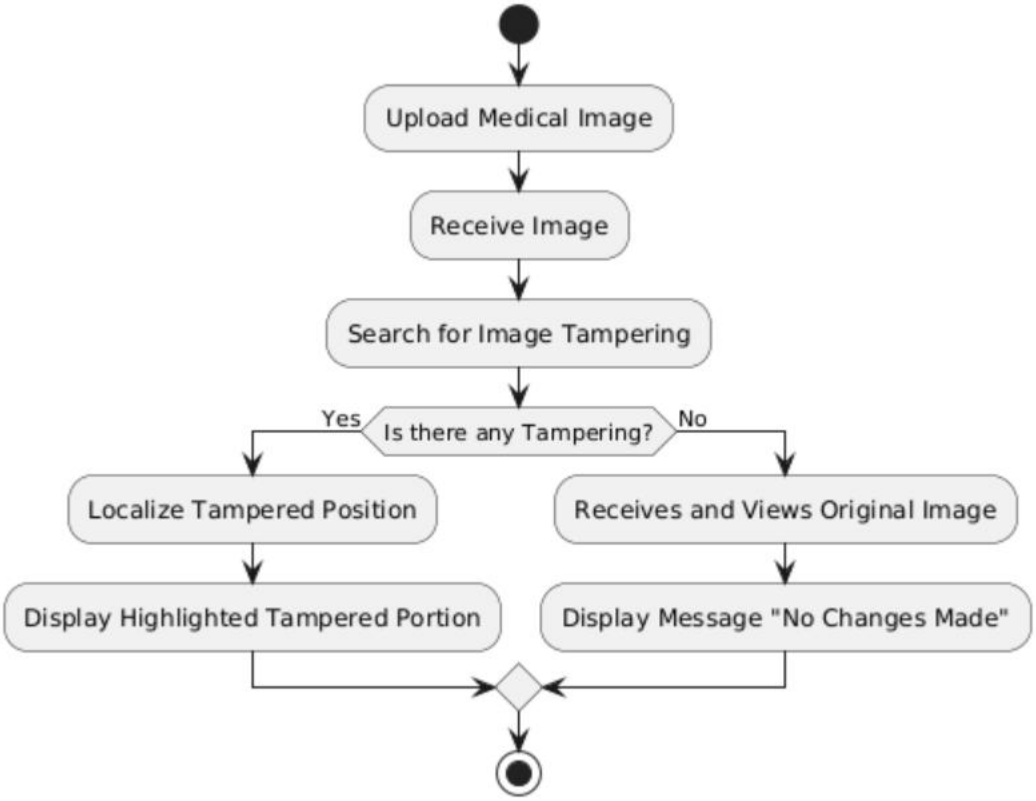

Upload Medical Image
Receive Image
Search for Image Tampering
Yes
Is there any Tampering?
No
Localize Tampered Position
Receives and Views Original Image
Display Highlighted Tampered Portion
Display Message "No Changes Made"

Fig 4. Activity Diagram

5. Work Breakdown Structure (WBS):

1. Research & Problem Definition
   1.1 Study medical image tampering
   1.2 Define project scope and datasets
   1.3 Establish evaluation criteria

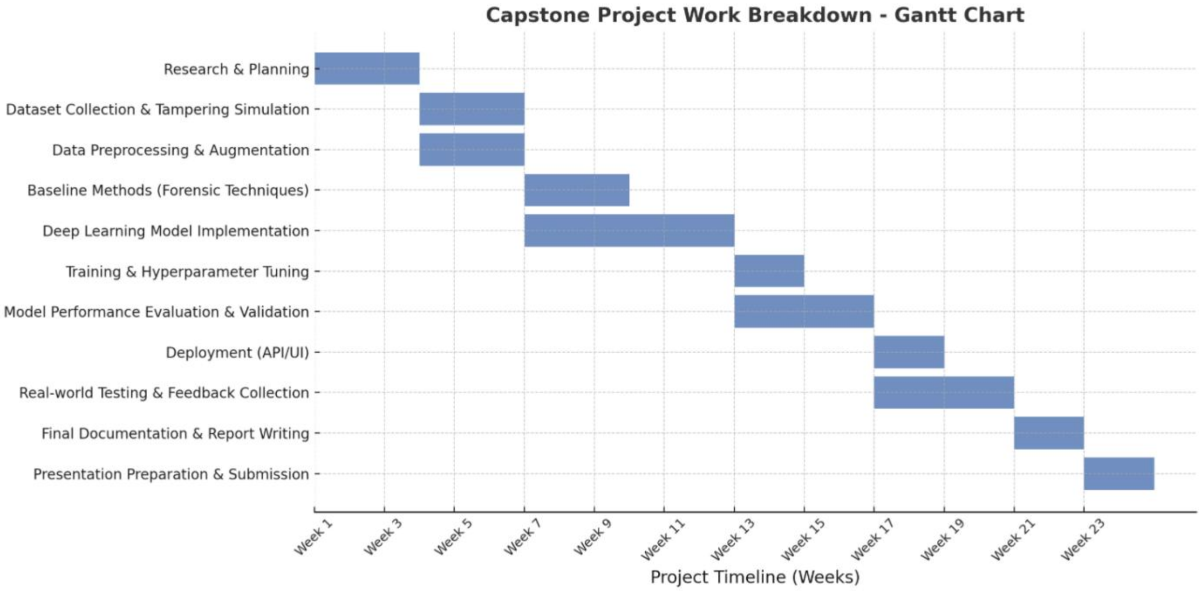
2. Data Collection & Preprocessing
   2.1 Collect datasets (original & tampered)
   2.2 Normalize and enhance images
   2.3 Annotate tampered and genuine images

3. Model Development
   3.1 Implement forensic baseline techniques
   3.2 Train deep learning models (CNNs, Transformers)
   3.3 Extract explainable features

4. Model Training & Evaluation
   4.1 Train with various architectures
   4.2 Evaluate and tune hyperparameters
   4.3 Compare against existing methods

5. Deployment & Testing
   5.1 Develop API or Web Interface
   5.2 Conduct real-world testing
   5.3 Prepare final documentation and research report

6. Scheduling



7. Specific Requirements

7.1 Functional Requirements

7.1.1 Image Acquisition & Preprocessing

• FR1: The system shall support image formats including DICOM, PNG, JPEG, and TIFF.

• FR2: The system shall allow users to upload medical images for analysis.

• FR3: The system shall preprocess images by resizing, noise reduction, and normalization to maintain uniform quality.

7.1.2 Tampering Detection & Localization

• FR4: The system shall analyze an image to detect any modifications or tampering.

• FR5: The system shall classify the type of tampering (e.g., splicing, copy-move, enhancement manipulation).

• FR6: The system shall highlight tampered regions using bounding boxes or heatmaps.

7.1.3 Machine Learning & Deep Learning Implementation

• FR7: The system shall extract image features using statistical analysis and deep learning models.

• FR8: The system shall utilize CNN-based models for automated tampering detection.

• FR9: The system shall provide an explainability feature to show how the AI model made a decision.

7.1.4 Performance Evaluation & Reporting

• FR10: The system shall generate a detailed tampering report indicating detected inconsistencies.

• FR11: The system shall allow manual verification and user feedback integration.

7.2 Non-Functional Requirements

7.2.1 Performance Requirements

• NFR1: The system shall analyze an image in less than 5 seconds.

• NFR2: The tampering detection algorithm shall have an accuracy of at least 95%.

7.2.2 Security Requirements

• NFR3: The system shall implement end-to-end encryption for secure medical image storage.

• NFR4: The system shall comply with HIPAA and GDPR regulations for medical data privacy.

7.2.3 Usability Requirements

• NFR5: The system shall have a user-friendly web interface for medical professionals.

• NFR6: The system shall provide interpretable results with visual overlays on tampered images.

7.2.4 Scalability & Compatibility

- NFR7: The system shall support multiple imaging modalities (MRI, CT, X-ray, mammogram).
- NFR8: The system shall be deployable on cloud-based and on-premise servers.