DocSentinel: An AI-Based Document Revision Tracker for Automated, Auditable, and Semantic Change

Problem-9 - Document Revision Tracker

Name: Dhruv Mishra
Affiliation: Department of Biosciences and Bioengineering, Indian Institute of Technology Guwahati
Email ID: m.dhruv@iitg.ac.in
Mobile Number: +91-8770674766

Abstract

Organizations generate and update thousands of documents, policies, contracts, technical reports, and SOPs that must be reviewed and verified for compliance. Manual comparison between versions is slow, inconsistent, and vulnerable to human error. **DocSentinel** is a comprehensive, AI-driven **Document Revision Tracker** that automatically identifies, classifies, and summarizes differences between two document versions across formats such as PDF, Word, Excel, and scanned images. The system combines **layout-aware text understanding**, **semantic language models**, and **visual change-detection CNNs** to detect textual, numerical, and graphical modifications with high precision. It outputs an **interactive diff viewer** and a **digitally signed audit log**, ensuring transparency and traceability. Lightweight model variants (LayoutLMv3-Tiny, SBERT-MiniLM) and on-premise execution guarantee feasibility and data privacy. DocSentinel's design directly addresses industrial requirements for reliability, regulatory compliance, and large-scale deployment, enabling faster and error-free document verification. DocSentinel uniquely integrates visual, structural, and semantic intelligence into a single verifiable document-comparison framework, an advancement beyond current enterprise tools.

Introduction and Foreground

Document revision tracking is central to industries where precision and accountability matter pharmaceuticals, law, finance, and manufacturing. Yet, most organizations still rely on manual or text-only comparison tools that fail when documents contain images, tables, or

paraphrased text. Existing utilities such as Adobe Acrobat Compare and Draftable detect only literal textual differences and are ineffective on scanned or layout-rich documents.

Recent advances in **Document AI** (e.g., LayoutLM, Donut, DiT) and **semantic embeddings** (SBERT) have made it possible to analyze both the *content* and *structure* of documents. In parallel, **Siamese convolutional networks** have shown strong results in change-detection tasks, distinguishing true alterations from visual noise. Despite these developments, no unified framework yet provides an explainable, multi-modal, and auditable document comparison system that integrates textual, visual, and structural understanding. **DocSentinel** bridges this gap by combining layout-aware feature extraction, semantic similarity modeling, and CNN-based image comparison to deliver reliable revision tracking suitable for enterprise workflows.

Solution

1. System Architecture

DocSentinel's pipeline consists of five stages:

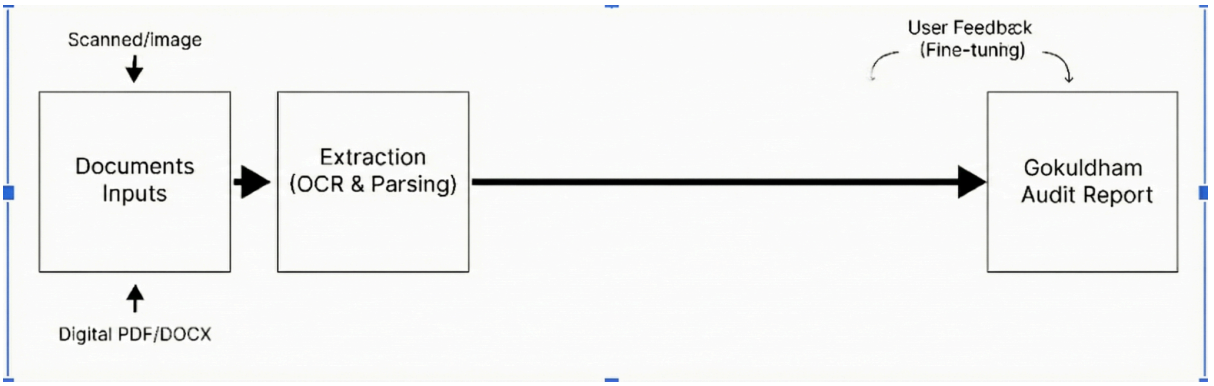
1. **Ingestion & Normalization**
    - Accepts PDFs, DOCX, XLSX, and image files.
      Converts all documents into a unified intermediate representation containing extracted text, layout coordinates, and page images.
    - Tools: PyMuPDF, PDFPlumber, Tesseract/PaddleOCR (for OCR), OpenPyXL (for spreadsheets).
2. **Feature Representation**
    - **Textual Stream:** Sentence embeddings via SBERT-MiniLM to capture meaning and detect paraphrases.
    - **Layout Stream:** Spatial and structural features via LayoutLMv3-Tiny or Donut-Small models.
    - **Visual Stream:** CNN (Arch-conv5, late fusion Siamese) extracting semantic visual features for detecting figure or stamp alterations.

3. **Change Detection Modules**

- **Structural Diff:** Aligns sections, paragraphs, tables, and figures using layout and text signatures.
- **Semantic Diff:** Computes cosine similarity between embeddings to identify contextual or numerical modifications.
- **Visual Diff:** Employs a change-detection CNN trained to localize altered regions while filtering scanning noise.
- **Hybrid Fusion:** Combines pixel-level (early fusion) and semantic (late fusion) cues to ensure both sensitivity and robustness.

4. **Summarization & Reporting**
   - Natural-language summaries of detected edits (e.g "Table 2: Cell B4 changed from ₹ 12,000 to ₹ 15,000").
   - Presents side-by-side diff visualization and produces a **tamper-evident audit log** (SHA-256 hash + digital signature).
5. **Security & Integration**
   - All computation is performed locally ; no external data transfer.
   - Modular APIs allow integration with enterprise (SharePoint, Alfresco, etc.).
   - Lightweight models and ONNX Runtime enable near real-time inference on standard hardware.

2. Workflow Overview



(Figure 1: DocSentinel workflow diagram — Document Inputs → Extraction → Multi-Modal Analysis → Change Fusion → Audit Report)

3. Challenges and Mitigation - Hybrid Fusion Sensitivity

The most significant technical challenge in DocSentinel is the **Hybrid Fusion** module that merges pixel-level visual cues with semantic text understanding. Visual differencing is extremely sensitive minor layout shifts or margin adjustments may appear as changes while semantic models may overlook subtle yet critical wording differences.To mitigate this, DocSentinel adopts a **two-stage filtering strategy**:

1. **Low-level Filtering:** Employs **Structural Similarity Index (SSIM)** and **perceptual hashing (pHash)** to remove cosmetic or rendering differences before semantic analysis.
2. **Adaptive Confidence Fusion:** Each candidate change receives two confidence scores visual and semantic. A weighted scoring scheme emphasizes changes validated by both modalities.
3. **Feedback Tuning:** User review feedback refines thresholds iteratively, increasing false positives over time.

   Visual Diff + Semantic Diff → Weighted Fusion → Last Decision

This design preserves **high recall for genuine content changes** while maintaining **clarity and precision** in the reviewer's interface.

Implementation Plan

Stage	Tasks / Activities	Tools / Technologies	Expected Deliverables
Data Preparation	Collect sample documents; create synthetic pairs with controlled edits.	Python, DocLayNet, ICDAR datasets	Evaluation dataset
Model Integration	Implement OCR + layout parsing; fine-tune LayoutLMv3-Tiny and SBERT-MiniLM; integrate CNN diff module.	PyTorch, Hugging Face, ONNX Runtime	Multi-modal comparison engine
Fusion & Reporting	Develop rule-based fusion; generate human-readable summaries; design audit schema.	FastAPI, pandas, cryptography libs	Change log & viewer
Interface & Testing	Build a web dashboard; user validation with precision/recall metrics.	React, PDF.js, Docker	Usable prototype with metrics

Hardware / Software Requirements: Standard x86 system (8 GB RAM CPU or GPU optional), Python 3.10+, PyTorch 2.0+, Docker.

Evaluation Metrics: Precision, Recall, F1-score for change detection; Intersection-over-Union for visual bounding boxes; time reduction factor for human review.

Quantifiable Outcomes

- ≥ 90 % recall on meaningful changes and ≤ 10 % false positives.
- 4–5× faster revision cycles than manual review.
- Generation of tamper-evident audit reports with traceable change history.

Industrial and Societal Impact

- Ensures regulatory and quality compliance in domains requiring strict documentation control.
- Reduces financial and legal risk associated with untracked document modifications.
- Promotes trustworthy digital governance through explainable AI and secure audit mechanisms.
- Scalable architecture enables integration into enterprise content management systems and cloud deployment pipelines.

References

1. Pun, A. K., Javed, M., & Doermann, D. (2023). *A Survey on Change Detection Techniques in Document Images.* IEEE Access.
2. Zanella, R., et al. (2018). *Spot the Difference by Object Detection.* arXiv:1810.10329.
3. Zhang, Z. (2022). *Layout-Aware Information Extraction for Document-Level Tasks.* ACL Proceedings.
4. Microsoft Research. (2020). *Semantic Table Structure Identification in Spreadsheets.* Technical Report.
5. Zhang, Y. (2020). *Learning to Detect Table Clones in Spreadsheets (LTC).* Microsoft Research.
6. Shafique, A. (2022). *Deep Learning for Change Detection: A Comprehensive Review.* Remote Sensing Letters.
7. Hunt, J. W. (1976). *An Algorithm for Differential File Comparison.* Communications of the ACM, 19(6), 418–422.

8. Huridocs. (2021). *PDF Document Layout Analysis Toolkit.* GitHub Repository.
9. DocLayNet Dataset. (2022). *Large-Scale Layout Understanding Dataset.* Hugging Face Hub.
10. ICDAR Change Detection Dataset. (2019). *Benchmark for Scanned Document Comparison.*
11. NAVER AI Lab. (2022). *Donut: Document Understanding Transformer.*
12. Xu, Y., et al. (2021). *LayoutLMv3: Pre-Training for Document AI.* arXiv:2111.08245.
13. Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence Embeddings Using Siamese Networks.* EMNLP Proceedings.
14. Camelot & OpenPyXL Libraries. (2023). *Python Packages for Table Extraction and Spreadsheet Parsing.*
15. PDFPlumber & PyMuPDF Libraries. (2023). *Structured Text and Layout Retrieval Tools.*
16. Tesseract OCR & PaddleOCR. (2023). *Optical Character Recognition Engines.*
17. ONNX Runtime. (2023). *Open Neural Network Inference Framework.* Microsoft Corp.
18. Wang, Z., Bovik, A. C., et al. (2002). *Image Quality Assessment: From Error Visibility to Structural Similarity (SSIM).* IEEE Transactions on Image Processing.