IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 6, June 2025

DOI: 10.17148/IJARCCE.2025.14615

Unmonitored Legacy Data Identification

Rutuja Karkande[1], Vaishali Kharade[2], Pranali Sonawane[3], Pratiksha Taral[4], Prof. Dr. N. A. Mulla[5]

B.E. Information Technology, Bharati Vidyapeeth's College of Engineering for Women, Pune, India[1-4]

Information Technology, Bharati Vidyapeeth's College of Engineering for Women, Pune, India[5]

Abstract: Legacy systems often suffer from a lack of clear ownership of code files and documents, leading to significant challenges in maintenance, security, and operational stability. Without defined accountability, it becomes difficult to track changes, address bugs, and implement necessary updates, increasing the risk of security vulnerabilities due to outdated or deprecated components. This project introduces a systematic approach for identifying unmonitored legacy data, categorizing unmaintained files, and establishing an efficient management framework. The proposed solution leverages automated techniques to analyze file metadata, assess modification patterns, and determine ownership attribution. Additionally, the system generates reports on unmaintained code and unmanaged documents, facilitating informed decision-making for risk mitigation. By implementing structured file ownership and maintenance protocols, organizations can enhance system security, improve operational efficiency, and ensure long-term software sustainability.

Keywords: Unmonitored, Legacy Data, Metadata, Unmaintained Files.

I. INTRODUCTION

Many organizations still rely on legacy applications—older software systems that are essential for business operations. However, as these systems age, they become increasingly difficult to maintain due to issues such as outdated code, missing documentation, and unclear ownership of files. Often, the original developers are no longer available, or the software was developed by third-party vendors, making it hard to track changes and updates. Without proper maintenance, these systems can lead to high operational costs, security vulnerabilities, and reduced efficiency.

One of the biggest challenges with legacy systems is the presence of unmonitored or outdated files. Over time, files such as old scripts, configuration files, and unused documents accumulate, making the system cluttered and difficult to manage. Since there is no clear ownership of these files, it becomes challenging to determine which files are still necessary and which can be archived or removed. This lack of accountability can lead to serious risks, including security breaches, compliance issues, and unexpected system failures.

To address these challenges, this project aims to develop an automated tool for managing legacy applications. The tool will scan software repositories, analyze file metadata, and track modification history to identify file ownership and maintenance status. It will also detect outdated and unmaintained files that may pose risks to the system. By automating these tasks, the tool will help organizations reduce maintenance costs, improve security, and ensure that their software systems remain stable and efficient.

Furthermore, implementing an automated system for legacy data management will enable organizations to establish a structured approach to handling outdated files. This will improve accountability by assigning ownership to specific files, ensuring regular updates, and reducing like likelihood of system failures due to outdated components.

II. LITERATURE REVIEW

In the paper *Official Document Identification and Data Extraction using Templates and OCR*, Cosmin Irimia, Florin Harbuzariu, Ionuț Hazi, and Adrian Iftene present a system designed to transform scanned images of official documents into structured textual data. The primary motivation behind this approach is to streamline bureaucratic processes and facilitate the secure digital sharing of personal documents. A notable use case highlighted in the paper involves a citizen scanning their identity card and sending it via email to a police institution with the subject "criminal record." Based on preconfigured email rules, the system automatically downloads the attachment, extracts the necessary personal data, and generates the corresponding criminal record, which is then sent back to the requester via email. This system represents a practical and privacy-conscious approach to document processing. Unlike many commercial solutions that require cloud connectivity, this system operates entirely offline and is open-source, ensuring both transparency and data security—an important factor when handling sensitive personal information. The authors also emphasize extensibility through a template-based design. For each new document type, a single representative template is sufficient to allow the system to identify and process that document class. This template-driven architecture makes the solution adaptable across a range

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 6, June 2025
DOI: 10.17148/IJARCCE.2025.14615

of administrative document formats, provided they follow relatively consistent layouts. Despite these advantages, the system's reliance on predefined templates poses limitations in more dynamic or unstructured data environments. While the template model proves effective for known and structured documents—such as ID cards—it may not scale efficiently in cases involving diverse, outdated, or poorly catalogued legacy data, where document formats are often unknown or vary significantly over time. This challenge is particularly relevant in the broader context of unmonitored legacy data identification, where systems must be capable of inferring structure without prior templates. The authors also address image quality issues, recognizing that many scanned documents, particularly in legacy archives, may be degraded or low-resolution. To mitigate this, their system employs a range of preprocessing techniques prior to the OCR phase, thereby improving the reliability of text extraction under suboptimal conditions. Overall, the work of Irimia et al. provides a strong foundation for structured document recognition in well-defined use cases. However, for projects focused on large-scale legacy data identification, their approach underscores both the value of template-driven design and the limitations such designs encounter in uncontrolled or heterogeneous document collections. [2] The paper presents the design and development of a Document Scanner Application using Python, aimed at simplifying the scanning, storage, and management of documents. The application allows users to scan documents and convert them into clear, editable, and storable formats such as images or PDFs. The core idea is to let users select specific portions of a document to scan, providing clarity and flexibility in output. The system is built using Python libraries and performs several tasks including image acquisition, edge detection (using the Canny algorithm), contour detection, and perspective transformation to enhance scan quality. It also includes text recognition features to extract and save text from scanned images. The app uses a GUI for selecting files and supports operations like saving images, converting them into text, and storing output efficiently. The methodology consists of five key steps: selecting the image, scanning it through edge detection and transformation, saving the scanned image, recognizing text through OCR, and saving the output text. The app aims to provide a user-friendly, ad-free, and secure scanning environment, ensuring mobility, reliability, and backup support. In conclusion, the application effectively addresses issues of document digitization, saving time and manual effort while offering flexibility in scanning and format.

III. METHODOLOGY

3.1 System Architecture:

Expanding on the architecture, the system begins when user uploads or selects a document from the repository. This document may be a technical report, code file, or any unstructured data that needs to be processed for ownership identification. The input provided by the user acts as the starting point of the system's workflow, triggering subsequent stages of analysis and data extraction.

Once the document is provided, it undergoes a repository scanning process. This step involves reading the file, checking its structure, and preparing it for further processing. The system may identify file types, encoding formats, and other structural attributes to ensure compatibility with the subsequent metadata and content analysis modules. The output of this step is a scanned version of the file ready for analysis. Following repository scanning, the scanned document is sent to the Metadata Extraction module. This component is responsible for extracting details such as the file's author, creation and modification dates, file size and other embedded information. Metadata provides essential clues for document ownership and is valuable for organizing and filtering documents during reporting. Once the metadata is extract, all the extracted metadata is forwarded to the database for storage. This database serves as a reference for identifying and verifying ownership based on document content and employee details. It ensures that ownership detection is not based solely on content, but also supported by verified employee information which enhances the accuracy of the identification process. The scanned file is simultaneously processed by a powerful Natural Language Processing (NLP) and Deep Learning module. This component parses the document's content to extract meaningful information such as contributor names, writing style patterns and contextual references. It uses advanced machine learning techniques to infer ownership by matching text patterns with employee's profiles.

After processing the document through NLP, deep learning and metadata extraction the system consolidates all findings into a structured report. This report includes key details such as identified owner of the document, relevant content summary, extracted metadata and flags for outdated or unmaintained documents.

3.2 Process Flow:

1. **User Input (Document Repository)**: The process begins when a user uploads or selects a document from the repository. This document may be a technical report, code file, or any unstructured data that needs to be processed for ownership identification. The input provided by the user acts as the starting point of the system's workflow, triggering subsequent stages of analysis and data extraction.

2. **Repository Scanning**: Once the document is provided, it undergoes a repository scanning process. This step

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 6, June 2025

DOI: 10.17148/IJARCCE.2025.14615

involves reading the file, checking its structure, and preparing it for further processing. The system may identify file types, encoding formats, and other structural attributes to ensure compatibility with the subsequent metadata and content analysis modules. The output of this step is a scanned version of the file ready for analysis.

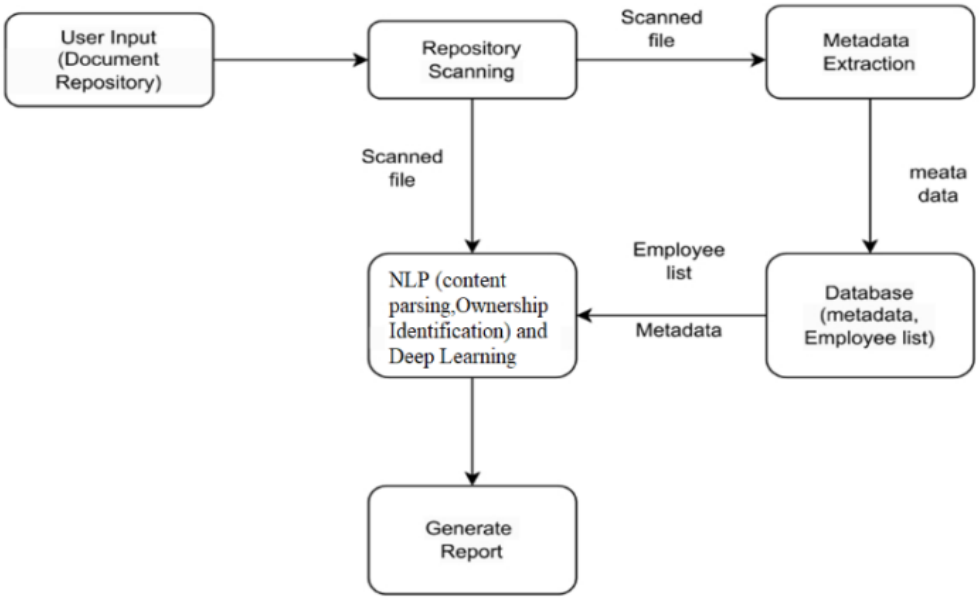

Fig 1. System Architecture

3. Metadata Extraction: The scanned document is then sent to the Metadata Extraction module. This component is responsible for extracting details such as the file's author (if mentioned), creation and modification dates, file size, version history, and other embedded information. Metadata provides essential clues for document ownership and is valuable for organizing and filtering documents during reporting. The extracted metadata is forwarded to the database for storage.

4. Database (Metadata, Employee List): All extracted metadata, along with an existing list of employees, is stored in a centralized database. This database serves as a reference for identifying and verifying ownership based on document content and employee details. It repository. This document may be a technical report, code file, or any unstructured data that needs to be processed for ownership identification. The input provided by the user acts as the starting point of the system's workflow, triggering subsequent stages of analysis and data extraction. ensures that ownership detection is not based solely on content, but also supported by verified employee information like names, roles, and departments, which enhances the accuracy of the identification process.

5. NLP (Content Parsing, Ownership Identification) and Deep Learning: The scanned file is simultaneously processed by a powerful Natural Language Processing (NLP) and Deep Learning module. This component parses the document's content to extract meaningful information such as contributor names, writing style patterns, and contextual references. It uses advanced machine learning techniques to infer ownership by matching text patterns with employee profiles. This step plays a crucial role in identifying relevant content and attributing authorship or responsibility, even when metadata is incomplete or missing.

6. Employee List to NLP Module: The Employee List stored in the database is also utilized by the NLP module. It helps in validating and strengthening the ownership identification process by comparing document content with known employee information. For example, if a name or email ID appears in the text, the system can match it against the list and improve confidence in the identified ownership. This cross-verification ensures that ownership suggestions are more accurate and trustworthy.

7. Generate Report: After processing the document through NLP, deep learning, and metadata extraction, the system consolidates all findings into a structured report. This report includes key details such as the identified owner of the document, relevant content summary, extracted metadata, and flags for outdated or unmaintained documents. The final report serves as a useful tool for organizations to manage legacy data, assign responsibility, and take informed decisions regarding document retention, migration, or deletion.

8. RESULT: The implemented system effectively achieved the core objectives of automatic document ownership identification and legacy data detection within organizational repositories. Through the Natural Language

DOI: 10.17148/IJARCCE.2025.14615

Processing (NLP) techniques, the system was able to:

Identify Document Ownership: The system successfully matched documents and code files to their rightful owners by analyzing metadata such as file names, author information , and modification history. It achieved a high accuracy rate linking files to active employees using employee dataset comparisons and NLP-driven text analysis.

Detect Unmaintained Documents and Code: Files that had not been modified within a pre-defined time threshold were automatically flagged as outdated or unmaintained. This helped in recognizing legacy files that might pose security or compliance risks.

Generate Metadata Reports: After scanning repositories, the system produced detailed reports that included file type, ownership, ownership status, created date ,last modified date, and outdated or unmonitored files. These reports were displayed through a user-friendly web interface, enabling easier auditing, documentation, and cleanup activities.

The experimental evaluation showed that the system could handle large repositories with diverse file types efficiently, providing consistent and reliable outputs. The NLP for contextual analysis proved effective in managing unstructured and semi-structured data formats. The system's modular design also supports future scalability and integration with existing enterprise data governance frameworks.

IV. CONCLUSION

This research successfully demonstrates the development of an intelligent system designed to automatically scan and analyze repositories to identify document ownership and detect legacy or unmaintained data. By integrating using Natural Language Processing (NLP) techniques for metadata analysis, the system achieves accurate identification of document ownership and authenticity. The core modules—Document Ownership Identification, Unmaintained Code and Document Detection, and Metadata Report Generation—work collaboratively to enhance data governance and repository transparency. The system flags outdated or unmanaged files, allowing organizations to mitigate risks associated with orphaned or obsolete data. Furthermore, the automated report generation feature provides a structured and actionable overview of the repository, supporting documentation, auditing, and cleanup processes. This contributes to improved operational efficiency, regulatory compliance. The promising results of this study suggest that NLP can be effectively applied to repository management and digital file governance. Future work may focus on expanding the model's adaptability to various data types and environments, integrating real-time monitoring, and enhancing the system's accuracy.

REFERENCES

[1]. Cosmin Irimia, Florin Harbuzariu, Ionut Hazi, Adrian Iftene, "Official Document Identification and Data Extraction using Templates and OCR" 26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022)

[2]. Sachin Parihar, S Ayushman, Rohan Deshpande , Shantanu Pal, Sourabh Yadav , Prof. Akshita Sharma: "DOCUMENT SCANNER APPLICATION USING PYTHON"

[3]. Eman Alajrami , Belal A. M. Ashqar , Bassem S. Abu-Nasser, Ahmed J. Khalil , Musleh M. Musleh , Alaa M. Barhoom , Samy S. Abu-Naser "Handwritten Signature Verification using Deep Learning"

[4]. A. Amari, M. Makni, W. Fnaich, A. Lahmar, F. Koubaa, O. Charrad, M. A. Zormati, and R. Y. Douss, "An Efficient Deep Learning-Based Approach to Automating Invoice Document Validation," arXiv preprint arXiv:2503.12267, Mar. 2025.

[5]. M. Chauhan, A. Satbhai, M. A. Hashemi, M. B. Ali, B. Ramamurthy, M. Gao, S. Lyu, and S. Srihari, "Vision-Language Model Based Handwriting Verification," arXiv preprint arXiv:2407.21788, Jul. 2024.

[6]. Cosmin Irimia, Florin Harbuzariu, Ionut Hazi, Adrian Iftene, "Official Document Identification and Data Extraction using Templates and OCR" 26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022)

[7]. Sachin Parihar, S Ayushman, Rohan Deshpande , Shantanu Pal, Sourabh Yadav , Prof. Akshita Sharma: "DOCUMENT SCANNER APPLICATION USING PYTHON"

[8]. Eman Alajrami , Belal A. M. Ashqar , Bassem S. Abu-Nasser, Ahmed J. Khalil , Musleh M. Musleh , Alaa M. Barhoom , Samy S. Abu-Naser "Handwritten Signature Verification using Deep Learning"

[9]. A. Amari, M. Makni, W. Fnaich, A. Lahmar, F. Koubaa, O. Charrad, M. A. Zormati, and R. Y. Douss, "An Efficient Deep Learning-Based Approach to Automating Invoice Document Validation," arXiv preprint arXiv:2503.12267, Mar. 2025.

[10]. M. Chauhan, A. Satbhai, M. A. Hashemi, M. B. Ali, B. Ramamurthy, M. Gao, S. Lyu, and S. Srihari, "Vision-Language Model Based Handwriting Verification," arXiv preprint arXiv:2407.21788, Jul. 2024.

