BFSI- OCR OF BANK STATEMENTS

Milestone 1 Report

PREPARED BY:
Jasneet Arora

Table of Contents

1. Introduction

The BFSI (Banking, Financial Services, and Insurance) OCR (Optical Character Recognition) project aims to develop an automated system for extracting and analysing data from financial documents using OCR technology, Azure AI Services, and OpenAI's Large Language Models. This first milestone focuses on the critical phase of establishing data infrastructure and preparation pipeline. The successful implementation of this foundation is crucial for the project's overall success as it ensures high-quality input data for the OCR processing stages that will follow.

2. Objectives

The primary objectives for Milestone 1 were to establish reliable data collection methods, implement robust quality control processes, set up secure cloud storage infrastructure, and develop an efficient metadata management system. These objectives were designed to create a scalable and maintainable foundation for the project's subsequent phases.

3. Methodology

The project implementation followed a systematic approach to ensure comprehensive data management and quality control. The methodology began with automated web scraping for initial data collection, followed by a multi-stage quality control process. This was complemented by cloud-based storage implementation and thorough metadata extraction and management procedures. Each stage was designed to maintain data integrity while optimizing processing efficiency.

4. Technical Implementation

4.1 Data Collection System

The data collection phase utilizes the Bing Image Downloader for automated acquisition of financial document images. The system is configured to collect five main categories of documents: bank statements, cheques, profit and loss statements, salary slips and transaction history. Python libraries os and shutil are employed for efficient file management and organization. This automated approach significantly reduces the time required for data collection while maintaining consistent document categorization.

4.2 Quality Control Framework

Quality control via PIL and OpenCV ensures image integrity. PIL checks resolution, sharpness, and format. OpenCV detects distortions, noise, and blurriness. Automated screening removes substandard images, followed by manual verification to maintain high quality.

4.3 Cloud Infrastructure

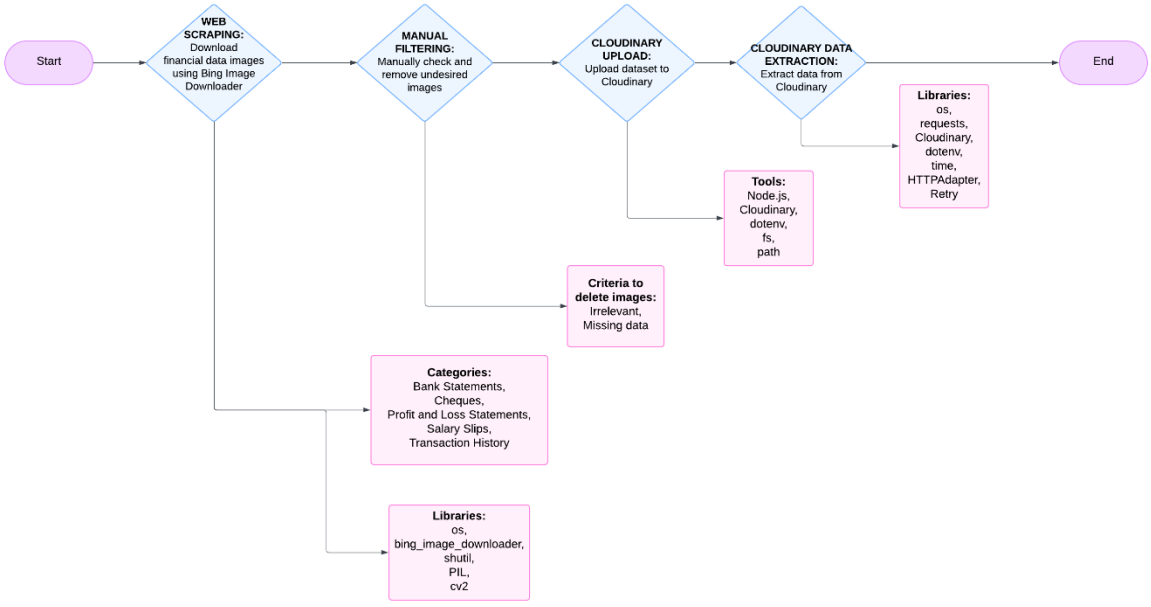
Cloud storage implementation utilizes Cloudinary as the primary platform for scalable and efficient media storage, supported by Node.js modules fs and path for local file operations. The dotenv library manages environment variables securely, ensuring sensitive credentials remain protected. The system organizes storage categorically to enable streamlined file management and reliable data retrieval.

4.4 Data Extraction

The metadata management system leverages the Cloudinary API for efficient data extraction and organization. The requests module, enhanced with HTTPAdapter and Retry, ensures reliable API calls by handling transient errors effectively. dotenv securely manages environment variables, while os facilitates system-level operations and time aids in process timing. This robust setup ensures proper categorization, streamlined accessibility, and maintained data integrity.

5. Results and Deliverables

5.1 Pipeline Implementation



The implemented pipeline represents a comprehensive workflow from initial data collection through final storage and organization. Each stage is designed for optimal efficiency while maintaining strict quality standards.

5.2 Key Achievements

The milestone has successfully delivered a fully functional data preparation pipeline, established quality control mechanisms, and implemented secure cloud storage solutions. The system is now ready for integration with OCR processing components in subsequent phases.

6. Challenges and Solutions

Throughout the implementation, several significant challenges were encountered and addressed:

6.1 Image Quality

Challenge: A portion of the downloaded images were of poor quality or irrelevant.

Solution: A multi-stage manual filtering process was implemented to retain only high-quality, relevant images for OCR processing.

6.2 Free Database for Storing Image Data

Challenge: Finding a free, scalable storage solution capable of handling a large volume of image files.

Solution: Cloudinary's free tier was selected, providing adequate storage capacity and image management features to meet the project's needs.

6.3 Process Automation

Challenge: Automating the data processing pipeline while maintaining quality control.

Solution: A semi-automated workflow was developed, balancing efficiency with necessary quality control measures to ensure data integrity.

7. Next Steps

The next phase of the project will focus on OCR integration, including dataset preparation for OCR processing, test case development, and validation framework creation. System enhancements will include storage optimization, automation process improvements, and enhanced data categorization capabilities.

8. Appendices

Appendix A: Technical Architecture

The pipeline architecture consists of four main stages:

The Web Scraping Stage initiates the process using Bing Image Downloader for data collection. This is followed by the Manual Filtering Stage, where image quality and completeness are verified. The Cloudinary Upload Stage handles secure data transfer to cloud storage, and finally, the Data Extraction Stage manages metadata organization and accessibility.

Appendix B: Technical Specifications

The development environment utilizes Node.js for backend operations, essential Python libraries for data processing, and the Cloudinary platform for storage management. Quality control parameters include checking for blurred images and ensuring that all images are paired with the required data.