



UNIVERSITÀ DI PISA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Master's Degree in Artificial Intelligence and Data Engineering

FileMind

Matteo Comini

ACADEMIC YEAR 2023/2024

GitHub Repository: <https://github.com/comocomo39/BPG>

1. Introduction

1.1 Project Context and Motivation

In the modern business environment, the effective management of digital documents has become essential for operational efficiency and regulatory compliance. However, the growing volume of documents (such as identity cards, invoices, contracts, and receipts) makes manual management inefficient and prone to errors. To address these challenges, the digitization and automation of document processes represent a strategic solution. This project aims to develop an application that leverages advanced technologies like Optical Character Recognition (OCR) and generative artificial intelligence to automate document analysis, renaming, and organization.

The primary objective of this project is to create a system that uses AI to analyse digital documents, extract key information, and rename them based on their content. This automation not only enhances file organization but also reduces the risk of human errors. By integrating AI, the system can accurately identify and catalog complex documents, enabling companies to streamline repetitive workflows, save time, and improve operational efficiency.

1.2 Project Objectives

The project aims to develop an application capable of:

- Extracting information from documents through advanced OCR technologies such as Tesseract.
- Using Gemini's artificial intelligence for semantic analysis of documents and extraction of critical information.
- Automatically renaming documents based on their content, with consistent and predefined formats.
- Organizing renamed files into an intuitive folder structure, facilitating access and management.
- Integrating a simple and intuitive graphical interface to manage the document uploading, analysis, and renaming process.
- Creating JSON files that store the extracted data from documents, implementing an automatic protocol system that ensures file traceability.

This system stands out for its ability to adapt to different types of documents and for the integration of AI technologies that enhance the accuracy and speed of processing.

This introduction outlines the project context, highlighting its relevance in the modern business landscape and justifying the use of artificial intelligence to improve document management.

2. Technological Overview

2.1 System Description and Project Architecture

The application automates document management by combining Optical Character Recognition (OCR) and generative artificial intelligence (AI). It handles various file types, including text documents (.txt, .pdf) and images (.png, .jpg), extracting relevant information through advanced AI techniques. The core technologies include:

- **Tesseract OCR:** An open-source engine used to extract text from image files. It can process various image formats (e.g., PNG, JPG, TIFF) and recognizes text in different languages, enabling the analysis of scanned documents.
- **Google Gemini AI:** A generative AI platform used for the semantic analysis of extracted text. Gemini identifies document types (e.g., identity cards, invoices, contracts) and extracts structured information like names, invoice numbers, dates, etc. The system interacts with Gemini via an API, sending prompts to guide the analysis and extract specific data.
- **Tkinter:** A Python library for creating the graphical user interface (GUI), which allows users to upload documents, view processing statuses, and see the final results such as the new file name and storage path.
- **JSON:** Used for structured storage of the extracted information. The system generates JSON files to keep track of extracted data and ensures the automatic protocol registration of documents.

The project workflow involves several key steps:

1. **File Upload:** Users upload documents through a Tkinter-based interface. Files can be in various formats (images or text), and the system processes them immediately after upload.
2. **Text Extraction:** For image files, Tesseract OCR extracts visible text, while text files (.txt, .pdf) are read directly. The extracted text is sent to the AI module for further analysis.
3. **Semantic Analysis:** The extracted text is analysed by Gemini AI, which identifies the document type and extracts relevant data, returning a structured JSON output with key information (e.g., names, invoice numbers).

4. **File Renaming and Organization:** Based on the analysis, the system renames files using a predefined convention (e.g., cid_Name_Surname.jpg for identity cards, invoice_Number.jpg for invoices) and moves them to an appropriate folder structure organized by user and document type.
5. **JSON File Update:** The system updates corresponding JSON files (e.g., invoices.json, contracts.json) with the extracted data and assigns a progressive protocol number to ensure traceability.

This architecture efficiently manages the entire document workflow, integrating multiple technologies to achieve automated, accurate, and organized document processing

2.2 Main Features

In addition to the basic processing stages, the system offers several advanced features:

- **Automatic Protocol Registration:** Each document is assigned a protocol number that increases progressively based on the files present in the system. This allows precise tracking of the order in which documents are processed and maintains a chronological record.
- **Centralized User Data Management:** Information related to users (name, surname, invoice number, etc.) is stored in a structured JSON format, facilitating the search and consultation of data.
- **Advanced Graphical Interface:** The application uses Tkinter to present the user with a user-friendly interface that allows monitoring the status of each file and viewing the processing results in real time.

3. Application Details

3.1 Workflow

The user interface provides real-time feedback, displaying the following details for each file:

- **Original File Name:** The name of the file as uploaded by the user.
- **Processing Status:** Indicators such as "In progress," "Completed," or "Error" that reflect the current state of the file processing.
- **Final File Path:** The location where the renamed file has been saved.

- **New File Name:** The renamed version of the original file, based on the extracted content.

Once the file is uploaded, the system proceeds with text extraction (for image files) or reading the content (for text files). The extracted text is then passed to the AI module for semantic analysis, file renaming, and organized storage, ensuring efficient and accurate document management.

3.1.1 Text Extraction

For image files, the application uses Tesseract OCR to extract the text. This open-source OCR engine is configured to handle images in various formats (e.g., PNG, JPG) and supports multiple languages. The extracted text is then sent to the AI module for further analysis. For text files (.txt, .pdf), the content is read directly without the need for OCR.

3.1.2 Document Analysis with Gemini

After extracting the text, it is sent to the Gemini generative AI for analysis. Gemini is configured to perform semantic analysis of the text, automatically identifying the document type (e.g., identity card, invoice, contract, receipt) and extracting key information such as:

- Name and surname (for identity cards and contracts)
- Invoice number and issue date (for invoices)
- Payment date and amount (for receipts)
- Contract ID and duration (for contracts)

Gemini's analysis returns a JSON output containing all the key information identified in the document. This information is then used to determine the new file name and organize it in the appropriate folder.

3.1.3 Automatic File Renaming

Based on the document type and extracted information, the original file is automatically renamed according to a predefined convention:

- **Identity cards:** cid_Name_Surname.jpg
- **Invoices:** invoice_Number.txt
- **Contracts:** contract_Number.txt
- **Receipts:** receipt_Number.txt

For example, if a file contains an identity card for "Mario Rossi," the file will be renamed as cid_Mario_Rossi.jpg. For an invoice, the file will be renamed based on

the extracted invoice number (e.g., invoice_12345.txt). Automatic renaming helps streamline file management and makes future retrieval simpler.

3.1.4 File Organization in Folders

Once renamed, the file is moved to a specific folder within an organized structure by user and document type. The folder structure is as follows:

- documents/name_surname/cid_name_surname for identity cards
- documents/name_surname/invoice_number for invoices
- documents/name_surname/contract_number for contracts
- documents/name_surname/receipt_number for receipts

This hierarchical organization allows for clear and structured document management, making it easier to search and retrieve documents based on document type and user.

3.2 Document Improvement Feature

In addition to the basic operations of extraction and organization, the application offers an advanced feature that allows users to improve and correct the analysis results through the graphical interface. Specifically, users can view and manually modify the data extracted from the document in cases where errors or ambiguities occur. This feature ensures that users retain full control over the output, enabling corrections and adjustments where necessary.

The graphical interface includes a text field that displays the analysis results and allows the user to modify them before confirming the renamed file. This is particularly useful in cases where OCR or Gemini AI fails to interpret certain information correctly (e.g., handwritten text or low-quality documents).

3.3 Graphical Interface

The graphical interface is designed to be simple and intuitive, making the application easily usable even for users without advanced technical skills. The GUI is built using **Tkinter**, a Python library for creating graphical interfaces. The main features of the interface include:

- **File Table:** Displays details for each uploaded file, including the original file name, processing status, final file path, and the new file name assigned.
- **Progress Bar:** A progress bar shows the progress of the analysis process, providing real-time feedback to the user during file processing.
- **Upload Button:** Allows the user to select and upload files to be analysed, starting the entire workflow with a single click.

- **Status Messages and Alerts:** The interface provides visual and auditory alerts to notify the user when a file has been successfully analysed, renamed, and moved to the correct folder, or if an error occurred during processing.

Processamento Documenti			
Nome File	Stato	Dettagli	Nuovo Nome File
dsvdsvdsv.txt	Completato	./documenti/GIOVANNI_NERI\contratto_CT-8243.txt	contratto_CT-8243.txt
dsvsvsv.txt	Completato	./documenti/GIOVANNI_NERI\ricevuta_RC-2115.txt	ricevuta_RC-2115.txt
ppppppp.jpg	Completato	./documenti/PASQUALE_CROCCO\cid_PASQUALE_C	cid_PASQUALE_CROCCO.jpg
Immagine WhatsApp	Completato	./documenti/MATTEO_COMINI\cid_MATTEO_COMINI	cid_MATTEO_COMINI.jpg

Elaborazione completata.

Seleziona File

3.4 JSON File Management

One of the system's core components is the management of the extracted information, which is stored in JSON files to ensure structured and easily accessible data. Each document type (e.g., invoices, contracts, identity cards) has its corresponding JSON file that stores all the key information.

For example, the invoices.json file stores information such as the invoice number, customer name, invoice date, and total amount. These data are recorded along with the path of the renamed file and a progressive protocol number to track all documents in a clear and orderly manner.

Every time a new document is processed, the system:

1. Updates the appropriate JSON file with the extracted data.
2. Adds a progressive protocol number.
3. Saves the JSON file so that it is available for future consultation or integrations.

4. Integration with Gemini API

Integration with Google's Gemini API is one of the key aspects of the project, as it allows leveraging advanced generative artificial intelligence to analyse uploaded documents and extract specific information automatically. This approach enables

accurate and scalable management of large volumes of data, reducing the margin of error typical of manual tasks.

4.1 Model Configuration

Once the API key is obtained, the connection to the Gemini API is configured. The project uses the "gemini-1.5-flash" model to perform semantic analysis and generate content based on the uploaded documents.

The model configuration allows control over important aspects such as temperature, which influences the creativity of the AI's responses, and the maximum number of tokens generated (8192), which ensures handling of large documents.

```
import google.generativeai as genai

# Configure the Google Gemini API
genai.configure(api_key=API_KEY)

# Set up the model with specific parameters
generation_config = {
    "temperature": 1,
    "top_p": 0.95,
    "top_k": 64,
    "max_output_tokens": 8192,
    "response_mime_type": "text/plain",
}

model = genai.GenerativeModel(
    model_name="gemini-1.5-flash",
    generation_config=generation_config,
)
```

4.2 Analysis Prompt

Gemini's artificial intelligence uses advanced deep learning models designed to analyse text and images with a high level of accuracy. In the project, the Gemini API is used to analyse the text extracted from documents and return structured information. The logic behind the analysis is based on sending a prompt to the AI, which guides the model through the information extraction process.

The prompt used to interact with the AI is carefully structured to provide detailed instructions on how to analyse the document and what information to extract. For example, the prompt includes requests to identify the document type (e.g., identity card, invoice, contract) and to extract key data associated with each document type (e.g., name, surname, invoice number, date).

The prompt is tailored to the specific file type and guides Gemini in extracting the key information. This flexible structure allows the AI to handle various document formats.

```
def analyze_file_with_gemini(image=None, extracted_text=None, is_image=False):
    prompt = f"""
    Dear Gemini AI,

    Please carefully analyze the attached file and identify the following information:

    Extracted text from the image: {extracted_text}

    1. Document Type Identification:
    - If the file is an identity card, extract the following data:
      - First name
      - Last name
      - Date of birth (if present)
    - If the file is an invoice, extract the following data:
      - Invoice number
      - Issuer's name
      - Invoice date
      - Unit price
      - Quantity
    - If the file is a receipt, extract the following data:
      - Receipt number
      - Customer name
      - Payment date
      - Amount paid
    - If the file is a contract, extract the following data:
      - Contract ID
      - Employee name
      - Job title
      - Start date and contract duration
      - Annual salary

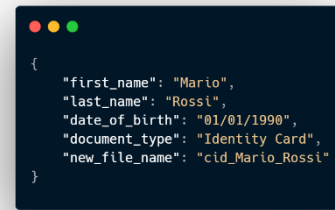
    2. File Renaming:
    - If the file is an identity card, rename the file in the format: 'cid_Firstname_Lastname'.
    - If the file is an invoice, rename the file in the format: 'invoice_number'.
    - If the file is a contract, rename the file in the format: 'contract_number'.
    - If the file is a receipt, rename the file in the format: 'receipt_number'.

    3. User Data Structure:
    - Return the user information (first name, last name, and other relevant data) in JSON format,
      including the new file name.

    Thank you for your assistance.
    """
```


4.3 Analysis Results

After sending the prompt to the Gemini AI, the model generates a response that includes the extracted information in JSON format. This JSON is then used to rename the file and organize it into predefined folders. Below is an example of the JSON result generated by the AI:



```
{
  "first_name": "Mario",
  "last_name": "Rossi",
  "date_of_birth": "01/01/1990",
  "document_type": "Identity Card",
  "new_file_name": "cid_Mario_Rossi"
}
```

The system processes this information to rename the file based on the document type and stores the extracted data in the corresponding JSON file. For example, an identity card will be renamed as cid_Mario_Rossi.pdf and archived in the correct folder.

4.4 Benefits of Integration

The integration with Gemini offers several benefits for the project:

- **Accuracy:** Thanks to the AI's ability to analyse and understand context, the extraction of data from documents is highly accurate.
- **Flexibility:** The model can adapt to various types of documents and formats, reducing the need for specific preprocessing.
- **Automation:** The process of renaming and organizing files occurs automatically, significantly reducing manual workload and the risk of errors.

5. Limitations, Potential Extensions, and Conclusion

5.1 System Limitations

The system developed offers significant benefits, but it has certain limitations that could be improved. Tesseract OCR struggles with handwritten text and complex document layouts, such as tables or images embedded within the text. Gemini AI, while effective, is constrained by token limits (8192 tokens per request), which may complicate handling large documents, and its performance depends on an active internet connection, which may limit its usability in offline environments.

Additionally, the system's performance with large files, such as high-resolution images or multi-page PDFs, could be affected, especially on devices with limited computational resources. This is particularly relevant when using OCR and AI to process large datasets.

5.2 Potential Extensions

Several improvements can be implemented to enhance the system:

- Support for additional document types, such as legal, financial, or medical documents, would increase its flexibility in various industries.
- Adding support for more file formats, such as Word documents, would allow better handling of complex documents often used in businesses.
- Enhancing the graphical interface (GUI) could make the system more user-friendly, including features like customizable themes, data editing tools, and search and filtering functionalities.

The structured JSON output of the system provides a clear advantage for future integrations. For instance, storing these JSON documents in a NoSQL database like MongoDB would facilitate fast querying and retrieval of document data. MongoDB's flexibility in handling large-scale datasets would also make it a powerful backend solution for managing growing volumes of documents, enabling future scalability improvements, including real-time processing and parallel task distribution.

Further integration with cloud services like Google Drive or Dropbox would enable automatic document uploads and synchronization, improving accessibility and collaboration. In addition, real-time processing and scalability enhancements, such as parallel processing or distributed computing architectures, would make the system more efficient in handling larger volumes of data.

5.3 Conclusion

The project demonstrates how combining Tesseract OCR with Gemini AI can significantly improve document management by automating the analysis, renaming, and organization of various document types. The system effectively reduces manual work, improves accuracy, and ensures a structured and traceable workflow.

By addressing the current limitations and implementing the proposed extensions, the system could become even more versatile and capable of handling larger volumes of data across multiple industries. Leveraging the JSON files for integration into a NoSQL database like MongoDB could open new possibilities for efficient data management, making the system a valuable tool for businesses seeking to automate their document processing workflows. Future developments would further optimize performance, enhance scalability, and improve the overall user experience.