**Technical Report – OCR & Summarization App**

**Public Web App Link** :- [**https://handwritten-text-ocr-summary-app-nbppx9hcd7evxau925sc3f.streamlit.app/**](https://handwritten-text-ocr-summary-app-nbppx9hcd7evxau925sc3f.streamlit.app/)

**GitHub Repository :-** [**https://github.com/CH9812/Handwritten-Text-OCR-Summary-App**](https://github.com/CH9812/Handwritten-Text-OCR-Summary-App)

**Mistral API Key** :- CjiUMO1FbQufCah3kenGUKVwSJDoS8Uh

**Reviewer Instructions to Test the App**

To evaluate the functionality of the OCR & Summarization App:

1. **Open the Streamlit App** (locally or deployed, if available).
2. When prompted, enter the following **Mistral API Key**:
3. Upload a **PDF or Image** file (preferably handwritten) or paste a URL.
4. Click **"Run OCR + Summarize"** to test:
   * OCR extraction using Mistral
   * Summary generation using Hugging Face model
5. Review:
   * OCR Text Output
   * Generated Summary
   * JSON/TXT Download Buttons

**1. Overall Approach & Tech Stack**

The objective of the project was to build a simple and effective web app that performs **OCR on handwritten text** (PDFs or images) and generates a **human-readable summary**.

**Tech Stack:**

* **Frontend/UI**: [Streamlit](https://streamlit.io/) – for building an interactive web interface with minimal code.
* **OCR Engine**: [Mistral OCR API] – for extracting text from both printed and handwritten documents/images.
* **Summarization Model**: Hugging Face transformers pipeline using t5-small – for summarizing extracted OCR content.
* **Backend Utilities**: Python libraries (base64, json, time, etc.) to handle file encoding, response parsing, and output formatting.

**2. Key Challenges & Solutions**

| **Challenge** | **Description** | **Solution** |
| --- | --- | --- |
| **Multi-page Document Handling** | Only the first page of PDFs was being processed. | Need to verify Mistral’s multi-page handling or implement manual page splitting. |
| **Handwritten Table Parsing** | Ensuring accurate OCR for handwritten tables. | Used high-resolution image uploads + Mistral’s powerful OCR to capture structure. |
| **Long Text Summarization** | Hugging Face models have token limits. | Split OCR text into 800-character chunks and summarized each, then concatenated. |
| **File Type Support** | Needed to support both PDFs and images from upload and URL. | Used Streamlit’s file\_uploader and base64 encoding for consistent input formatting. |
| **Summary Generation Issues** | Abstractive summarization was often inaccurate or too generic. | Used text chunking and t5-small, but output still lacked precision |

**3. Testing & Validation**

* **Functionality Test**: App tested with:
  + Single and multi-page PDFs.
  + JPEG/PNG images of handwritten notes.
  + Uploaded and URL-based files.
* **Validation**:
  + Verified OCR output manually for accuracy on multiple samples.
  + Compared summary quality with raw text to ensure coherence and relevance.
  + Verified proper download of generated JSON, TXT files for both OCR and summary.
* **Edge Cases Tested**:
  + Empty or unreadable text files.
  + Documents with very little text (less than 10 characters).
  + Long text documents exceeding summarizer limit.

### ****Notes & Observations****

* **Tried multiple summarization models** (e.g., bart-large, flan-t5), but they struggled with raw OCR output.
* **Tried AWS Textract**, but:
  + Accuracy was poor for handwritten inputs.
  + It incurred billing charges.
  + Ultimately not worth it for this use case.

Thank You

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |