Contribution File for Handwritten Text Recognition and Keyword Search Project

Project Title:
Handwritten Keyword or Phrase Search System & Handwritten Kannada Text Recognition

Contributors:

- **Vishal**
- **Sudheer**
- **Sriniketh**

Overview of the Project

The main goal of this project was to tackle two key challenges:

1. **Handwritten Keyword or Phrase Search:** Implementing a system capable of searching and locating specific words or phrases in a corpus of handwritten documents, making the search more efficient and useful for users.
2. **Handwritten Kannada Text Recognition:** Building a model that can effectively convert scanned images of handwritten Kannada text into machine-encoded text using Optical Character Recognition (OCR).

The project relied on Google Cloud's Vision API for OCR, image preprocessing techniques to improve accuracy, and an intuitive search system to query the recognized text. Below is the detailed breakdown of each contributor's work.

Vishal’s Contribution

Vishal played a critical role in the preprocessing and image enhancement stages of the project. His efforts were essential in ensuring that the images used for OCR were optimized for accurate text recognition.

Image Preprocessing for OCR

- **Image Contrast and Sharpness Enhancement:**
  Vishal worked extensively on improving the quality of the handwritten images before they were processed by the Google Vision API. By applying image contrast and sharpness adjustments, he ensured that the text was more visible and easier for the OCR engine to process. Specifically, he used Python's `PIL` library functions `ImageEnhance.Contrast` and `ImageEnhance.Sharpness` to enhance the scanned images, which helped clarify faint or blurry handwriting.
- **Grayscale Conversion and Noise Reduction:**
  To make the OCR process more efficient, Vishal also implemented grayscale conversion, reducing the complexities introduced by colored backgrounds or ink variations in the scanned documents. In addition, he applied Gaussian blur techniques using `ImageFilter.GaussianBlur` to remove noise from the images, which minimized distractions and further improved the OCR's accuracy.

Google Vision API Integration

- **Text Detection Using Vision API:**
  Vishal was responsible for integrating Google Cloud's Vision API into the system. By using the API's document text detection feature, he configured the system to recognize Kannada handwriting by adding `"language_hints": ["kn"]`, which helped the OCR engine better interpret handwritten Kannada text. This setup enabled the accurate extraction of Kannada text from scanned images and allowed for large documents to be processed seamlessly.

Sudheer's Contribution

Sudheer was instrumental in building the core search functionality and ensuring that users could interact with the recognized text corpus efficiently. His primary focus was on making the search system fast, user-friendly, and intuitive.

Handwritten Corpus Search System

- **Keyword and Phrase Search Implementation:**
  Sudheer developed the search system that allows users to query recognized handwritten text from the corpus. He wrote the `search_text` function, which uses Python's `re` module to perform case-insensitive searches across the entire corpus. This function identifies all occurrences of a user-specified word or phrase and returns a list of matches with their positions in the text. The search mechanism was designed to handle large corpora efficiently, allowing users to search through multiple documents at once.
- **User Interface for Search Results:**
  Sudheer also focused on the presentation of search results within the Streamlit app. For

each match found in the corpus, he implemented a system that displays the context of the search query along with the filename where the match was found. To improve the user experience, he highlighted the searched keyword or phrase in red, making it easier for users to locate their search term within the document. He ensured that the results were displayed in an organized manner, even for large documents.

- **Search Result Visualization:**
  To provide users with insights into the distribution of search results across different documents, Sudheer implemented a visualization feature using Streamlit's bar chart functionality. This visual representation allows users to quickly see which documents contain the most occurrences of their search query, offering a document-wise breakdown of search hits.

Sriniketh's Contribution

Sriniketh primarily focused on integrating the full system and ensuring that the application was robust, user-friendly, and met the project's overall goals. His contribution revolved around putting together the different pieces and refining the end-to-end functionality.

System Integration and Corpus Management

- **File Uploader for Multiple Image Inputs:**
  Sriniketh took responsibility for enabling the user to upload multiple images to the system. Using Streamlit's `file_uploader`, he ensured that users could upload several JPG or PNG files in one go. This functionality was key in making the system scalable, as users often need to process large batches of handwritten documents.
- **Corpus Management and Saving Functionality:**
  Sriniketh was responsible for organizing and saving the recognized text in a structured way. After performing OCR on each uploaded document, he stored the recognized text in a dictionary (corpus) where each document's text was associated with its filename. He also implemented the functionality to compile the entire corpus into a single `.txt` file, allowing users to download all the recognized text in one place.

Performance Optimization and UI Improvements

- **OCR Performance Optimization:**
  Sriniketh worked on streamlining the process of sending images to the Google Vision API and processing the responses. He optimized the way images were converted into byte arrays for API processing, reducing the time it took to process each document and improving the overall performance of the application.
- **User-Friendly Interface for Corpus Display:**
  Sriniketh also contributed to the UI, particularly in how the recognized text was displayed. By using transparent containers with custom styling in Streamlit, he ensured

that the recognized text was easy to read, and users could preview the extracted content directly on the web interface. His use of Streamlit's `markdown` and `container` elements provided a clean, visually appealing way to present both the raw and processed text.

Conclusion

The combined efforts of Vishal, Sudheer, and Sriniketh led to a successful implementation of the Handwritten Keyword or Phrase Search System and the Handwritten Kannada Text Recognition model. The project tackled the challenges of image preprocessing, OCR accuracy, efficient keyword search, and intuitive user interfaces. The result was a scalable and user-friendly system that not only converts handwritten Kannada text into machine-readable text but also allows users to search through large volumes of handwritten documents quickly and easily.