Document Summary Assistant - Project Documentation

Project Overview
The **Document Summary Assistant** is a Python desktop application built using *Tkinter* that helps users extract text from PDF or image files, and generate concise summaries of the extracted text. It supports multiple summarization strategies: OpenAI GPT-3.5 (optional, requires an API key) Hugging Face Transformers pipeline Fallback simple heuristic summarization **Key Features**

- Extract text from PDF files (using *PyMuPDF (fitz)*)
- Extract text from images (using OCR with *pytesseract* and *PIL*)
- Summarize the text with OpenAI API, Hugging Face pipeline, or fallback method
- Simple GUI interface for ease of use

Required Modules
tkinter (standard library) fitz (PyMuPDF) pytesseract Pillow (PIL) transformers openai shutil, os, platform, threading (standard library) **Installation Instructions**
pip install pymupdf pytesseract pillow transformers openai
Note: Ensure Tesseract OCR Engine is installed and configured separately.

Main Components
1. Text Extraction:
- extract_text_from_pdf(path): Extracts text from PDF.
- extract_text_from_image(path): Extracts text from image.

2. Summarization:
- summarize_text(text, max_length, api_key): Uses OpenAI, Hugging Face, or fallback.

3. GUI (Tkinter):
- Load files, input OpenAI key, display text & summary, status updates.

Usage Example
1. Run the app.
2. Load a file.
3. Set max summary length & provide OpenAI key (optional).
4. Generate summary.

Fallback Behavior
Handles missing modules or API failures gracefully.

Important Notes
- Install Tesseract OCR separately.
- OpenAI API key is needed for OpenAI summarization.

Summary
Helps in summarizing large documents and integrates GUI, file handling, text extraction, APIs, and fallback logic.