**PROJECT OVERVIEW AND OBJECTIVES**

**Project Name:** DOCSee

**Researcher:** Feyza SAPAN

**Date:** 03.08.2025

**Institution:** BMS Defence Technologies Inc.

**Objective of the Project:**  
The primary aim of this project is to develop a desktop application that enables the digitization of physical documents and the extraction of their textual content into editable formats. The application integrates image processing techniques with Optical Character Recognition (OCR) to offer a high-accuracy and user-friendly solution for converting printed documents into machine-readable formats. DOCSee allows users to automatically detect document content and export it as either PDF or plain text (TXT) files.

**Problem Statement and Rationale:**  
In contemporary work environments, document digitization remains a manual, time-consuming, and often expensive process. Particularly in archival, academic, or bureaucratic contexts, there is a critical need to convert physical paper-based content into digital form. Existing mobile-based scanner apps are often limited in OCR accuracy and flexibility, while professional hardware scanners can be costly and inaccessible. DOCSee addresses this gap by offering a cost-effective, open-source, and platform-independent solution built using Python and its powerful ecosystem. It supports both image uploads and live webcam scanning, ensuring flexibility and adaptability to different user needs.

**Target User Groups:**

* Academics, researchers, and students
* Government and private sector employees handling official documents
* Individuals engaged in personal or institutional archiving
* General users seeking to digitize various types of documents

1. **TECHNOLOGIES USED**

Modern Python technologies and open-source libraries were employed in the development of this application. The selected tools cover both user interface (UI) design and advanced image processing tasks.

* **Python (3.10+)**  
  The primary programming language of the project. It was chosen due to its rapid development capabilities, extensive package ecosystem, and strong community support.
* **PyQt5**  
  Used for designing the graphical user interface. Core interface components such as windows, buttons, image displays, and dialog boxes were implemented using PyQt5.
* **OpenCV**  
  A robust computer vision library utilized for detecting document edges, enhancing image quality, and performing perspective transformations.
* **EasyOCR**  
  A deep learning-based OCR engine built on PyTorch, used for extracting text from documents. It supports multiple languages and provides high recognition accuracy.
* **PIL (Pillow)**  
  Applied for converting processed images into PDF files.
* **NumPy**  
  Used for performing matrix operations and transformations on image pixel data, especially in preprocessing steps.
* **datetime**  
  Utilized to generate timestamped filenames for automatically saving extracted content.

1. **FUNCTIONAL ARCHITECTURE AND USER WORKLOW**

The DOCSee application integrates several key functionalities into a seamless user experience by combining image processing, graphical interface design, and real-time document recognition.

**Image Acquisition Methods:**

Users can either upload scanned or photographed documents from their local file system or capture new images using the system’s webcam. Both methods are initiated from the main menu and direct the user to the corresponding processing interface.

**Automatic Document Detection:**

Upon image acquisition, the system automatically detects document edges using OpenCV. Rectangular contours are analyzed, and the most probable document area is extracted through adaptive thresholding, contour detection, and edge filtering.

**Manual Selection Tools:**

In situations where automatic detection fails or the detected area is inaccurate, DOCSee allows the user to define the document region manually. This can be done by selecting four corners or drawing a bounding box, both of which are handled within the same interface using interactive mouse input.

**Perspective Correction:**

All selected documents—either automatically or manually—undergo geometric transformation to correct skew and angle distortions. This perspective transformation ensures that the document is rectified to a flat, readable format suitable for OCR.

**Real-Time OCR Execution:**

Once the document area is finalized, the system triggers OCR in the background using a separate thread. EasyOCR is applied to extract text from the corrected image. Prior to recognition, preprocessing steps such as grayscale conversion, CLAHE (Contrast Limited Adaptive Histogram Equalization), and adaptive thresholding are used. Regular expressions are applied post-recognition to filter out noise and irrelevant data.

**Result Display and Export Options:**

The recognized text is shown in a popup editor with a side-by-side preview of the scanned image. Users can edit the text in place, then export it either as a .txt file or generate a multi-page .pdf document from all scanned pages.

1. **CHALLANGES ENCOUNTERED AND SOLUTIONS**

**Delay in OCR Processing:**

Initially, the OCR process caused noticeable delays, especially for larger or multiple-page documents. To resolve this, the OCR workflow was moved into a dedicated QThread, enabling the operation to execute in the background without interrupting the user interface.

**Freezing or Unresponsive User Interface:**

When performing heavy image operations such as perspective correction or thresholding, the GUI would occasionally freeze. This issue was mitigated by offloading all computationally intensive tasks to worker threads, maintaining a responsive and fluid user experience. A progress dialog was also introduced to indicate the status of long-running tasks.

**Low OCR Accuracy on Raw Input:**

In early tests, OCR output often included irrelevant or poorly recognized characters due to image quality issues. To improve accuracy, preprocessing steps such as grayscale conversion, CLAHE (Contrast Limited Adaptive Histogram Equalization), and adaptive thresholding were applied. Additionally, post-processing with regular expressions was implemented to eliminate empty or noisy OCR results.

**Limited Manual Control over Document Area Selection:**

Users encountered difficulties when automatic detection failed or misidentified document boundaries. To address this, a manual selection tool was developed, enabling users to either select four corner points or draw a rectangle around the document. This ensured more accurate and user-guided cropping.

1. **CONCLUSION AND FUTURE WORK**

The DOCSee application has been successfully completed as a stable and flexible desktop solution for document digitization using OCR technology. With its user-friendly interface, support for multiple documents, and export functionality in both PDF and TXT formats, it meets the core requirements of various use cases. The ability to switch between automatic and manual detection methods makes it adaptable in scenarios where document structure varies.

The following improvements are proposed for future development:

* Adding support for multilingual OCR (e.g., Turkish, German, French)
* Integrating a built-in spellchecking module to refine extracted text
* Developing a mobile version for Android and iOS platforms
* Enhancing the OCR pipeline to detect tables, figures, and layout structures

These enhancements would further improve DOCSee’s usability and expand its range of applications in academic, administrative, and commercial settings.