Department of Computer Science & Engineering
The University of Texas at Arlington

Architectural Design Specification
CSE 4316: Senior Design I
Spring 2021

Optical Profilers
Document Classifier

Muhammad Daud
Xavier Wells
Bishwamitra Sapkota
Koshish Khadka

Revision History

Revision	Date	Author(s)	Description
0.1	4/1/2021	XW	document creation
0.2	4/11/2021	XW,BS,KK,DM	First Draft

Contents

List of Figures

List of Tables

1 Introduction

The Document Classifier system is a document identification solution. A user will be able to upload a file for identification, and Document Classifier will able to classify the document as recognized or unrecognized. A recognized document will also be scanned for a signature within a specified subsection of the recognized document. The system will also store the results of classifications to be retrieved by privileged users later. Document Classifier will be used as a proof of concept for State Farm. As a proof of concept, Document Classifier will be able to recognize five different document classes and detect signatures on these recognized documents. Document Classifier in its currently planned state is not intended to be available publicly or commercially. Document Classifier is designed specifically for State Farm, but could possibly be of value for any corporation that has to process large volumes of electronically scanned documents.

The Document Classifier system is made up of several separate components. The front-end of the system will consist of a simple web application. This web application will facilitate all user interaction with the system. The web application will be able send requests to upload documents and query for previous classification results. A RESTful API will facilitate all requests between the front-end and the back-end. Uploaded documents will be stored in a secured storage solution along with previous classification results. After uploading documents, they will be preprocessed to improve classification accuracy. The text will then be extracted from the documents and used to classify them. If the document is of a recognized type, then it will be scanned for a signature. The system is only expecting pictures or pdf scans of documents as input from the user. The user will then receive the classification result of the system as output. The original file will be stored on the server. The user will also be able to query for classification results and the original documents. Data will exist in other forms in intermediate classification steps, but will not be stored. The user will primarily just see the web interface for Document Classifier. Document Classifier is not intended to have any administrator or maintainer in the long-term, but if the system did, they would also see the storage solution and the classification model. The storage solution will be a database that will hold files and classification results. The classification result will include all preprocessing and text mining components. The uploaded documents will be stored in the database along with the classification results.

State Farm receives thousands of documents from its customers on a daily basis. The existing process of manual classification and verification of the documents not only costs a lot of time and money to the company but also is prone to errors. Having a system that can automatically classify and verify the documents on a certain basis, can save time, reduce operational cost and moreover, help the company provide faster and better service to its customers. In addition to saving time and money, this process will certainly provide faster and better experience to the customers and help improve the agent-customer interaction

2 SYSTEM OVERVIEW

The identiDoc system overview consists of two major components â the front end and the back end. These two components are highly independent of each other pass a minimal amount of data back and forth between each other.

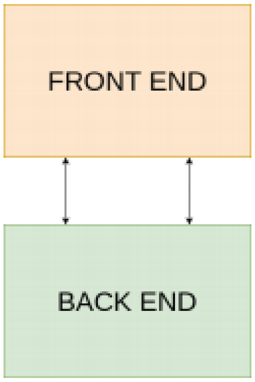

Figure 1: Optical Profilers System Overview

2.1 FRONT END DESCRIPTION

The front end, which consists primarily of our user interface (UI), handles file uploads and user queries. All interactions with the user are handled through the UI. All user interactions will return a response with a status. For example, the user will be notified of the document classification result.

2.2 BACK END DESCRIPTION

The back end consists of the API, storage solution, and the classification model. The API functions as a connection between the front end and the back end. The storage solution stores the files that are uploaded by the user and maintains a database of previous classification results. The classification model processes and classifies uploaded documents.

3 Subsystem Definitions & Data Flow

The user interface is the singular interface of the front end of the system. The user interface has two interior components â the query handling call requests a query of previous classification results from the API and the file handling call uploads a user-selected file for classification.

The back end of the system is comprised of three subsystems - the API, the storage solution, and the classification model. The API handles all interactions between the front end and the back end of the identiDoc system. The storage solution stores all of the static data, including the original uploaded files and records of classification. The classification model will classify the file.

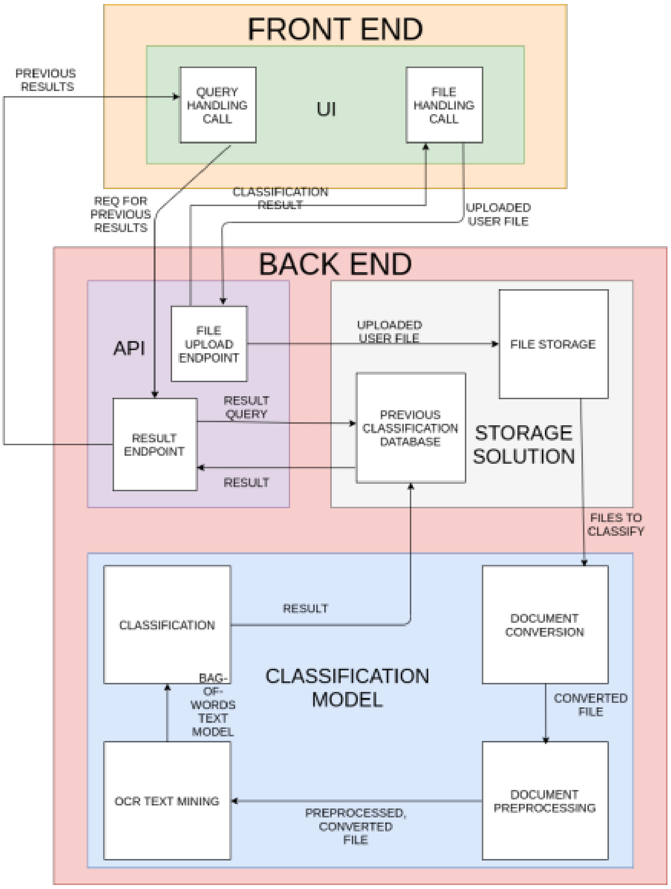

Figure 2: data flow diagram

4 Front End

4.1 USER INTERFACE (UI)

The UI will be the primary way in which the user will interact with the identiDoc system. The user will be able to upload a file for classification and retrieve previous classification results by classification date.

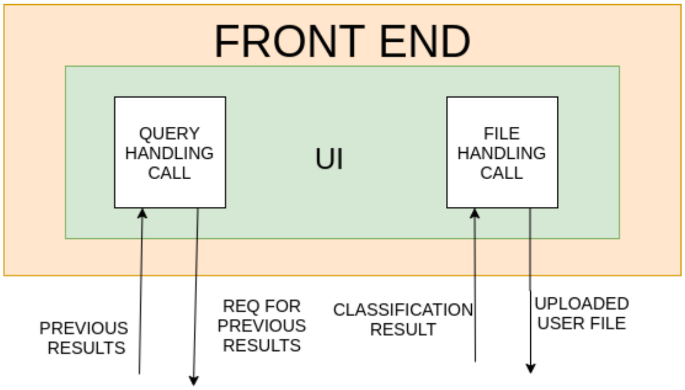

Figure 3: Front End Subsystem Description Diagram

4.1.1 Assumptions

The UI should be fully functional on any modern web browser. The user should be able to select any local file for classification by the identiDoc system. The user should be able to filter classification results by date of classification. All requests should be promptly returned from the back end (within two seconds).

4.1.2 Responsibilities

The UI is responsible for all user interactions with the identiDoc system. The UI should be able to successfully upload a file for classification and show either the classification result or an appropriate error message. The user should be able to select a date to query for classification results and display the results in an appropriate table.

4.1.3 Subsystem Interfaces

Table 2: User Interface Subsystem Interfaces

ID	Description	Inputs	Outputs
#01	Query Handling Call	Results query of the	HTTP GET request with selected date for query
#02	File Handling Call	File classification result	HTTP POST request with appended file to upload

5 Back End

5.1 API

The API will serve as the access point of the application. The API will handle all requests from the Front End of the identiDoc system. The user, through different API calls, will be able to upload files for classification and query for previous classification results.

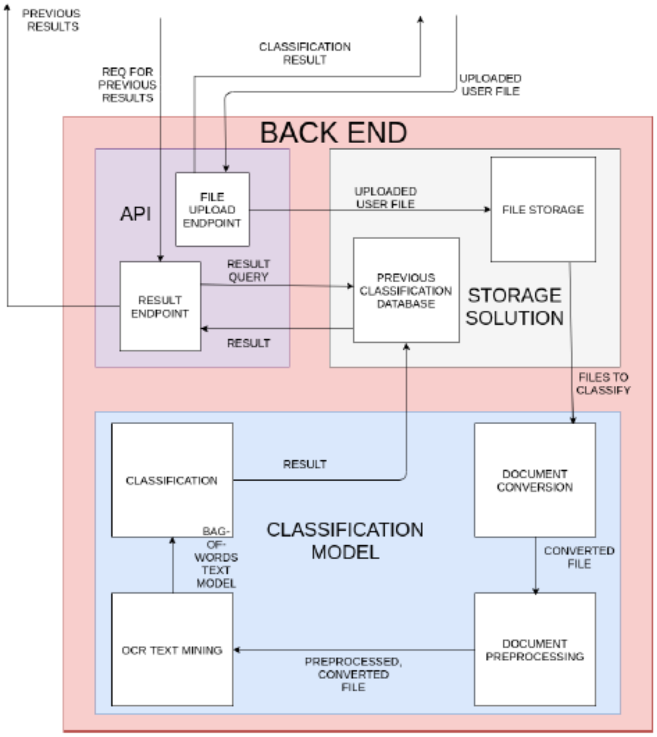

Figure 4: Example subsystem description diagram

5.1.1 Assumptions

The user should only be allowed to upload .pdf, .png, .jpg, .jpeg, or .txt files through the API. Each individual file should not exceed 24 MB. Only one file will be accepted for upload at a time. The user will only be able to query previous classification results by date. Currently, the official identiDoc web application is the only planned user interface to interact with the identiDoc API.

5.1.2 Responsibilities

The API is responsible for listening for a call from the UI requesting a query. The API then sends a request to the Storage Solution to query for the specified data. The API then returns the data that was retrieved. The API is also responsible for handling file uploads. The API retrieves the data from the UI and copies it to the specified File Storage Location. After uploading the file, it will be classified. The result will be given to the API to be sent back to the UI.

5.1.3 Subsystem Interfaces

Table 3: API Subsystem interfaces

ID	Description	Inputs	Outputs
#03	File Upload Endpoint	HTTP POST request with appended file	JSON response with classification result. User file for storage.
#04	Result Endpoint	HTTP GET request with date appended to URL. Results from a previous query.	Query for previous results. JSON response with query results.

5.2 Storage Solution

The Storage Solution will store all static data needed for the identiDoc system. This will include a database of records of classification results, and the actual files that were classified.

5.2.1 Assumptions

The database will only need create and read operations (not update nor delete). For time being, we are also assuming that we will have adequate space to store our uploaded documents on a server.

5.2.2 Responsibilities

The storage solution is responsible for storing all of the static data. The File Storage will receive the file from the API and save it. The classification sequence will also be triggered once the file is uploaded. After the classification is complete, the database will be updated with the result. The API can also ask the database for previous classification results. The database will execute the query and return the data.

5.2.3 Subsystem Interfaces

Table 4: Storage Subsystem Interfaces

ID	Description	Inputs	Outputs
#05	File Storage	Uploaded User File	File to be classified
#06	Previous Classifications Database	Result Query. Result from classification	Classification Result Previous Classification Result

5.3 Classification Model

The classification model is responsible for classifying documents.

5.3.1 Assumptions

All files that are accepted by the storage solution should be able to be classified.

5.3.2 Responsibilities

The classification model will be responsible for classifying the uploaded document. The original document will be converted to a standard file type (probably .png) and then preprocessed in preparation to be read by OCR technology. An OCR engine will read the file and create a bag-of-words model. The text can then be classified using a similarity score for classification. A similarity threshold will be set to deal with unrecognized documents. The classification model will pass off the resulting classification result to the storage solution.

5.3.3 Subsystem Interfaces

Table 5: Classification Model Subsystem Interfaces

ID	Description	Inputs	Outputs
#07	Document Conversion	Original File	Standard converted file
#08	Document Preprocessing	Standard converted file	Processed, converted file
#09	OCR Text Mining	Processed, converted file	Bag of words text model
#10	Classification	Bag of words text model	Classification result

REFERENCES