DOI: https://doi.org/10.61762/pjstvol7iss2art001 PLATFORM - A Journal of Science & Technology

Received: 26 August 2024, Revised: 11 December 2024, Accepted: 24 December 2024, Published: 31 December 2024, Publisher: UTP Press, Creative Commons: CC BY 4.0

BOOK CATALOGUING SYSTEM USING OPTICAL CHARACTER RECOGNITION AND NAMED ENTITY RECOGNITION

Shakirat Aderonke Salihu[1], Sodiq Olaniyi Bamidele-Alao[1], Adeyinka Tella[2], Hakeem Babalola Akande[3], Abdullateef Oluwagbemiga Balogun[4]*, Hammed Adeleye Mojeed[1,5], Fatima Enehezei Usman-Hamza[1] and Abimbola Ganiyat Akintola[1]

[1]Department of Computer Science, University of Ilorin, Nigeria
[2]Department of Library and Information Science, University of Ilorin, Nigeria
[3]Department of Telecommunication Science, University of Ilorin, Nigeria
[4]Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Malaysia
[1,5]Department of Applied Computer Science, Gdansk University of Technology, Poland

*E-mail: abdullateef.ob@utp.edu.my

ABSTRACT

Libraries function as knowledge preservers and agents of learning in communities. It is necessary to have access to well-organised books for efficient information retrieval and academic purposes. Nevertheless, traditional library systems' cataloguing books are often manual and thus laborious and inefficient, making them prone to errors. This calls for an automatic system for cataloguing books to streamline the exercise, enhancing accuracy. Therefore, this research proposes an automated book cataloguing system that combines both Optical Character Recognition (OCR) and Named Entity Recognition (NER). In particular, the Tesseract OCR Engine is employed to grab text from book images while spaCy identifies and classifies entities existing within the document. All these functionalities are supported by a Python-developed back-end via an effective API, and the front end operates on React.js, ensuring user-friendly interaction. The result of this research reduces the time consumed to create catalogue records for books in the library, making the process efficient and easier. The system was evaluated based on Word Error Rate (WER) and Character Error Rate (CER) for OCR, while NER components were evaluated using precision, recall, and F1-score. The results of OCR have an overall of 3.2% for CER and 5.7% for WER. The year as a component under NER has the highest precision of 99%, while ORG has the lowest precision of 91%.

Keywords: *Access control, convolutional neural network, face recognition, firebase, mobilefacenet*

INTRODUCTION

Library cataloguing is the process of creating and maintaining bibliographic and authority records in a library's system, and is a fundamental key to effective information management in libraries worldwide. As defined by Haider [1], it encompasses a diverse range of information materials, from books and serials to audio-visual media, cartographic resources, and digital files. The significance of cataloguing in library operations cannot be overstated, as it serves as the primary means of organising and generating metadata for vast collections of information materials. Without proper cataloguing, the sheer volume of resources in libraries globally would render access nearly impossible for users. Cataloguing and classification systems work in tandem to provide structured access to library materials and facilitate easy and efficient information retrieval. The evolution of cataloguing systems has seen three distinct eras: manual cataloguing, automated cataloguing featuring machine-readable cataloguing (MARC) standards and copy-cataloguing, and the current era of artificial intelligence (AI). This study focuses on this emerging AI era in library cataloguing, exploring its potential to revolutionise how libraries organise and provide access to their collections.

The AI era has revolutionised how we work and think [2]. A library collection that is organised according to standardised rules and practices should allow users

to retrieve information effectively, quickly, efficiently, and ultimately to maximise the usage of the collection anywhere in the world [3]. AI refers to the capability of digital computers, computer-controlled machines, or software to emulate intellectual characteristics typically associated with intelligent organisms, particularly humans. As defined by major AI scholars and textbooks, the field aims to design and develop "a fully conscious, intelligent, computer-based entity" with inherent advantages over humans in environmental perception and complex task optimisation [4]-[5]. AI systems are designed to record and analyse user actions, contributing to their learning and adaptation. The rapid advancement of science and technology has led to AI's integration across various aspects of life, enhancing human development and comfort [6]. The versatility of AI is evident in its wide-ranging applications across multiple industries. In healthcare, AI is being tested and deployed for tasks such as determining medication dosages, tailoring treatments to individual patients, and assisting in surgical procedures, demonstrating its potential to revolutionise numerous sectors beyond its original conceptualisation [7]-[8].

Machine learning (ML), a subset of AI, enables computer programs to learn and adapt to new data autonomously without human intervention. This capability is facilitated by deep learning (DL) algorithms that process vast amounts of unstructured data, including text, images, and video [9]. In Nigeria, academic librarians have recognised AI as a transformative force for developing intelligent libraries and have begun incorporating AI technologies into their systems to align with global trends in librarianship. However, despite these efforts, there is a notable lack of documentation regarding AI implementation in Nigerian academic libraries, as highlighted by Adejo and Misau [10]. The application of AI in libraries encompasses various technologies, including expert systems, natural language processing (NLP), DL, knowledge reasoning, and robotics, as identified by Barsha and Munshi [11]. While these AI applications are increasingly visible in libraries across developed and some developing countries, their adoption and impact in Nigerian educational institutions remain an area ripe for exploration and documentation.

The library's concern is efficiently and expeditiously providing access to its holdings. Thus, cataloguing activity is a bottleneck in making available newly acquired material to the patron and represents a major commitment of staff resources. This study investigates how to effectively use Optical Character Recognition (OCR) technology to extract digital text from scanned library resources. This approach aims to automate a tedious, usually manual task that is a part of library workflows. It offers a system that uses Named Entity Recognition (NER) capabilities to automatically identify key information items embedded within the extracted text, such as publication dates and author names. This effective extraction of crucial information creates a more effective way to catalogue books, which could significantly improve the precision and effectiveness of library cataloguing. Previously, cataloguing and classification were done manually, making the work very difficult, boring and time-consuming. Recently, most university libraries in Nigeria have joined their counterparts in advanced countries in using computers to process library collections [3]. There have been various eras of cataloguing and classification systems in libraries. These eras include the manual cataloguing and classification system, automated cataloguing and classification system using MARC and copy-cataloguing [12]. MARC was originally used to automate the creation of physical catalogue cards; now, the MARC computer files are accessed directly in the search process [3].

By leveraging OCR technology, the physical text of books can be converted into digital data, eliminating the need for manual data entry. This data can then be further processed through NER techniques. NER allows us to automatically identify key information within the text, such as author names, publication dates, and subject keywords. This combined approach of OCR and NER can significantly automate book cataloguing, leading to increased efficiency and accuracy. Furthermore, by extracting rich metadata through NER, libraries can create more comprehensive and searchable catalogues, ultimately enhancing the discoverability of their resources for patrons.

Summarily, the key contributions of this research are to:

1. Automate the process of book cataloguing using OCR and NER techniques.
2. Evaluate the performance of the proposed system and contrast it with human catalogers and cataloguing solutions from current studies.

RELATED WORKS

Humbel et al. [13] investigated the application of NER to early modern textual sources, primarily studied by Humanists. Their work highlighted a lack of detailed reporting on NER processes in Humanities disciplines, making finding relevant data and metrics in project outputs challenging. In a related study, Short [14] explored text mining applications in cataloguing by extracting text from digitised novels using OCR. This process yielded substantial amounts of text, but Short noted that the automated nature of OCR often resulted in various inaccuracies. The frequency of these errors varied significantly between books, influenced by factors such as typography, layout, and the item's physical condition. These studies collectively underscore the potential and challenges of applying advanced text-processing techniques to historical and literary materials in the context of library science and digital humanities.

Alghemei et al. [15] explore the integration of advanced image processing and NLP techniques to extract textual content from book cover images. Their approach leverages YOLOv8 for accurate text localisation, Pytesseract for OCR to digitise text regions, and spaCy for deep NLP analysis, aiming to detect text comprehensively. Faizullah et al. [16] surveyed OCR in Arabic, and their work explores the applications, techniques, and challenges of applying these NLP techniques in Arabic language text.

Kiessling et al. [17] introduced Kraken, an OCR system specifically designed for historical documents and rare scripts. It uses deep learning techniques to improve accuracy on challenging texts. NER for Book Metadata was developed by Tian et al. [18], which is a transformer-based NER model for extracting bibliographic information from book titles and subtitles. Their approach significantly improved identifying complex entity types like series names and edition information. Llabrés et al. [19] proposed a multimodal approach to book cataloguing that combines text and image analysis. Their system uses both cover images and OCR-extracted text to improve metadata extraction accuracy. Carbune et al. [20] presented a language-agnostic system for processing books in multiple languages. Their approach uses multilingual transformers to handle OCR post-processing and entity recognition across various languages.

Lafia et al. [21] developed a digitisation and processing workflow using a set of 25,744 semi-structured paper-based records from the G.I. Bill Mortgage administration from 1946 to 1954. The mortgagor's name and city, the mortgage amount, the Reconstruction Finance Corporation agent's location, one or more identification numbers, and the bank servicing the loan are all listed in these records. From these scanned historical records, the authors retrieved structured information to build a tabular data file and connect it to other reliable individual-level data sources.

The reviewed works focus solely on recognition and information extraction without addressing the step of image pre-processing. This proposed approach addresses this gap by incorporating image pre-processing techniques to enhance the overall text clarity of scanned library book images. This pre-processing step will reduce errors in the OCR engine's output, leading to more accurate data extraction for the library cataloguing process. Also, the datasets were locally sourced from a university to alleviate the time spent by their catalogers in book cataloguing. Furthermore, the reviewed works lack a user-friendly interface. This study addresses the need to deploy the system and ensure it has a visual interface that allows users to efficiently and easily interact with the system. This interface will provide:

1. Clear guidance and feedback on user inputs, ensuring data accuracy from the start.
2. Review and edit capabilities for the extracted information, allowing for human correction when necessary.

This study proposes an efficient and usable solution for automating library book cataloguing using OCR and NER technologies by incorporating image pre-processing and a user-friendly interface.

METHODOLOGY

This section presents and discusses the proposed study's research methodology and framework in detail, as presented in Figure 1.

Data Collection

A dataset of book images was collected from the University of Ilorin, Nigeria's library, to develop and evaluate the proposed automated book cataloguing

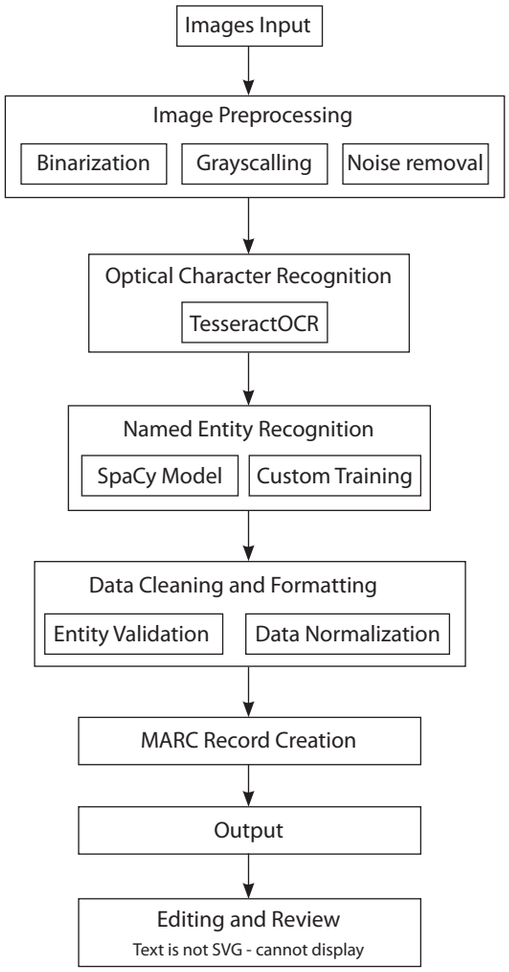


Figure 1 Framework of the proposed system

system. This includes capturing images of book covers, title pages, and other relevant sections using the CamScanner app installed on a smartphone. The CamScanner app allowed for convenient and efficient scanning of physical books by leveraging the phone's camera. The app's features, such as automatic edge detection, perspective correction, and image enhancement, helped to produce high-quality scans of the book pages and covers. Books were selected from various genres. This was done to ensure that the dataset represents a diverse range of book formats, layouts and fonts, thereby testing the robustness of the proposed system in handling different types of book metadata.

The scanning process involved capturing multiple images for each book, including the front cover, title page, publication details, and other sections deemed relevant for extracting bibliographic information. Personal information or sensitive content was obscured as necessary. The scanned book images were organised and labelled with relevant metadata, such as book titles and authors, to facilitate the NER model development's subsequent annotation and training phases. This dataset of scanned book images from the University of Ilorin library, captured using the CamScanner app on a smartphone, serves as the foundation for training and evaluating the OCR and NER components of the automated book cataloguing system. Table 1 provides a brief description of the dataset used for the study.

Table 1 Summary of the dataset collected

Discipline	No. of Books	No. of Images
Engineering	7	14
Mathematics	7	14
Physics	7	14
Chemistry	7	14
Total	28	56

Image Pre-processing

The scanned book images acquired from the data collection process passed through a series of pre-processing steps to enhance their quality and prepare them for the subsequent OCR stage. For this purpose, the OpenCV (Open-Source Computer Vision Library) was employed. OpenCV is a widely used open-source computer vision and ML library that provides a comprehensive set of tools and algorithms for image and video processing. Its cross-platform nature and extensive documentation make it a suitable choice for the image pre-processing tasks required in this study.

The specific pre-processing techniques applied using OpenCV include:

1. Image enhancement: Techniques such as contrast adjustment, brightness normalisation, and denoising are applied to improve the visual quality and clarity of the scanned book images.
2. Binarisation: Converting the grayscale or colour images into binary (black and white) format can enhance the performance of the OCR engine by reducing noise and improving text segmentation.
3. Skew correction: Any skew or rotation in the scanned images is corrected using OpenCV's geometric transformation functions, ensuring proper text alignment for accurate OCR.

4. Page layout analysis: OpenCV's connected component analysis and contour detection algorithms are utilised to segment the book images into text regions, separating the main content from headers, footers, and other non-textual elements.
5. Text region extraction: The identified text regions from the page layout analysis are cropped and extracted as separate images, facilitating focused OCR processing on the relevant textual content.

The pre-processed images, enhanced and optimised through OpenCV, were then fed into the OCR component of the automated book cataloguing system, improving the accuracy and reliability of the text recognition process.

Optical Character Recognition (OCR)

The open-source Tesseract OCR engine was utilised for the OCR component of the automated book cataloguing system. Tesseract is a highly accurate and widely adopted OCR engine developed by Google and now maintained by the Open-Source community. Its ability to recognise text in various fonts and styles is particularly beneficial for processing the varying typographic representations found in book materials. The pre-processed book images, obtained through the image pre-processing stage using OpenCV, served as input to the Tesseract OCR engine. Tesseract's powerful text recognition algorithms were employed to extract the textual content from these images accurately. Post-processing techniques, such as text cleanup and formatting, were applied to the OCR output to ensure consistency and conformity with the desired data formats required for subsequent stages of the book cataloguing process.

Named Entity Recognition (NER)

The spaCy library was employed for the NER component of the automated book cataloguing system. spaCy is a free, open-source, and highly efficient NLP library for Python designed to build production-ready applications. spaCy has powerful and accurate named entity recognition capabilities and customisation and transfer learning support. This flexibility allows for the development of a tailored NER model specifically adapted to the domain of book cataloguing.

To train the NER model, a dataset comprising book metadata and descriptions was collected and manually annotated with relevant entity types, such as author names, book titles, publishers, and publication dates. This annotated dataset served as the training data for the spaCy NER model. The trained spaCy NER model was integrated into the automated book cataloguing system, which processes the OCR-extracted text and identifies and classifies relevant entities, such as author names, book titles, and publishers. This extracted information will be utilised in the subsequent stages of the cataloguing process, including data integration and record matching.

By leveraging spaCy's powerful NER capabilities, the proposed system aims to achieve high accuracy in extracting and classifying bibliographic entities, a critical step in automating library book cataloguing workflow.

Training NER model

To train the spaCy NER model for the system, the dataset of library books collected was annotated with ground truth/reference data for named entities. The annotation process involves manually labelling named entities such as author names, book titles, and publishers in the textual data. This was done using a labelling tool or by directly manipulating the text files with appropriate markup.

Once the annotation is complete, the labelled data are then split into training and testing/validation datasets. The training subset was used to train the NER model, while the testing/validation dataset was utilised to evaluate the model's performance during and after the training process.

The spaCy library provides a powerful API and utilities for training custom NER models. The training process involves the following steps:

1. Load the training data into spaCy's data structures.
2. Initialise a new pipeline with the required components (e.g., transformer, entity recogniser).
3. Configure the training hyperparameters, such as batch size, number of iterations, and learning rates.
4. Start the training process by feeding the annotated data to the NER component for learning.
5. Periodically evaluate the model's performance on the validation subset during training.

6. Transfer learning or domain adaptation can be incorporated to enhance the model's performance.

The trained spaCy NER model will then be integrated into the automated book cataloguing system, identifying and classifying named entities from the OCR-extracted text. This will ensure accurate bibliographic data extraction, facilitating efficient cataloguing and metadata management.

Data Cleaning and Pre-processing

The MARC format is a widely adopted standard for encoding bibliographic data in library catalogues. To integrate the extracted bibliographic information into existing library systems, the back-end will implement a module dedicated to generating valid MARC records from the cleaned and formatted entity data.

The process of generating MARC records involves the following steps:

1. MARC Field Mapping: The back-end defines a mapping between the extracted entities (e.g., author names, book titles, publishers) and the corresponding MARC fields and subfields. This mapping will ensure that the extracted data is correctly positioned within the MARC record structure.
2. MARC Record Creation: Based on the defined field mapping, the back-end creates a new MARC record instance and populates the relevant fields and subfields with the extracted and cleaned data. This process will adhere to the MARC standards and guidelines, ensuring the integrity and validity of the generated records.
3. Handling Repeatable Fields: Some MARC fields, such as author names or subject headings, may be repeatable. The back-end implements logic to handle these repeatable fields, ensuring that multiple instances of the same entity type are correctly encoded within the MARC record.
4. Indicator and Subfield Management: MARC fields often have indicators and subfields that provide additional information or context. The back-end handles the appropriate assignment of indicator values and the correct placement of data within subfields based on the MARC specifications.
5. Record Validation: Before finalising the MARC record, the back-end performs validation checks to ensure compliance with MARC standards and guidelines. This may include checking for required fields, validating indicator values, and verifying the overall structure and integrity of the record.
6. Output Formatting: Once the MARC record is generated and validated, the back-end provides options to output the record in various formats, such as MARC-XML, MARC-21 binary, or other exchange formats supported by library systems.

The back-end leverages Python libraries and modules specifically designed for working with MARC records, such as PyMARC, to implement these functionalities. This library provides high-level interfaces for creating, manipulating, and validating MARC records and tools for reading and writing MARC data in different formats.

System Deployment

React.js was employed for the front-end development of the automated book cataloguing system. It serves as the primary technology for building the user interface components and interactions and is known for its popularity as an open-source JavaScript library maintained by Meta (formerly Facebook) and a large community of developers. In the context of the automated book cataloguing system, React.js and TailwindCSS were used to build the user interface components for data input, display, and interaction. This includes forms for entering or modifying bibliographic data, displaying data grids or lists, and implementing UI for navigation and user interaction.

React.js's modular and reusable component architecture was leveraged to create a maintainable and scalable codebase, enabling easy integration with other system parts, such as data storage and server-side communication. React's library ecosystem also provides access to various third-party packages and tools that can be utilised for specific functionalities, such as handling forms, managing state, and implementing user interface components and interactions.

Furthermore, Python was employed as the primary programming language for the back-end development and the creation of an Application Programming Interface (API) to handle the core logic and functionality of the automated book cataloguing system. The back-end was developed as a RESTful API (Representational State Transfer) using a Python web framework like Flask or Django. These frameworks provide a solid foundation

for building robust and scalable web applications, enabling the creation of a well-structured and modular codebase. The API will serve as the central hub for orchestrating the various components of the automated book cataloguing system, which includes:

1. Image Pre-processing: Exposing endpoints to handle the upload and pre-processing of book images using the OpenCV library.
2. OCR: Integrating the Tesseract OCR engine to perform text extraction from the pre-processed book images.
3. NER: Utilising the trained spaCy NER model to identify and classify entities such as author names, book titles, and publishers from the OCR-extracted text.
4. Data Integration and Matching: Implementing algorithms and logic for data cleaning, normalisation, record linkage, and entity resolution to match the extracted data with existing library catalogues or databases.

By leveraging Python's simplicity, versatility, and powerful ecosystem, the back-end development will entail the creation of a robust, scalable, and efficient API that serves as the backbone of the automated book cataloguing system, handling the core logic and functionality.

Performance Evaluation Metrics

To evaluate the accuracy and reliability of the OCR component, the following metrics will be employed:

1. Character Error Rate (CER): CER measures the ratio of total character-level mistakes in the OCR text compared to the total characters in the ground truth text. It indicates the overall character-level accuracy of the OCR engine. Lower CER values indicate higher accuracy.
2. Word Error Rate (WER): WER evaluates the accuracy of an OCR system at the word level by measuring the proportion of incorrectly recognised words relative to the reference text. Like CER, lower WER values signify better OCR performance.

On the other hand, NER component's performance will be assessed using the following metrics:

1. Precision: Precision measures the proportion of correctly identified entities out of all the entities predicted by the NER model. It quantifies the model's ability to avoid false positives.
2. Recall: Recall evaluates the proportion of correctly identified entities out of the total number of relevant entities present in the ground truth data. It reflects the model's capability to detect true positive instances.
3. F1-score: The F1-score combines precision and recall into a single metric, providing a balanced measure of the NER model's overall performance. It is calculated as the harmonic mean of precision and recall.

The OCR metrics (CER and WER) were computed by comparing the OCR output against the ground truth text for the testing subset. Similarly, the NER metrics (precision, recall, and F1-score) were calculated by comparing the predicted named entities against the ground truth annotations in the testing subset. By calculating these metrics, the performance of the OCR and NER components can be quantified and compared against benchmarks.

RESULTS AND DISCUSSIONS

Data Collection

A systematic data collection process was conducted at the Unilorin library to build the dataset, focusing on books from four key scientific disciplines: Engineering, mathematics, physics, and chemistry. Figure 2 shows samples of the raw images collected.

The data collection phase resulted in a dataset of 56 images representing 28 academic books across four scientific disciplines. This dataset provides a good foundation for developing and testing the system, particularly its performance in handling academic literature with varying layouts and content.

Image Pre-processing Results

The implementation of image pre-processing techniques using OpenCV significantly enhanced the quality of the scanned book images, leading to improved OCR performance. The following pre-processing steps were applied:

1. Grey scaling: Converting the colour images to grayscale simplified the image data while retaining essential text information. This step reduced the complexity of subsequent operations and improved processing speed.

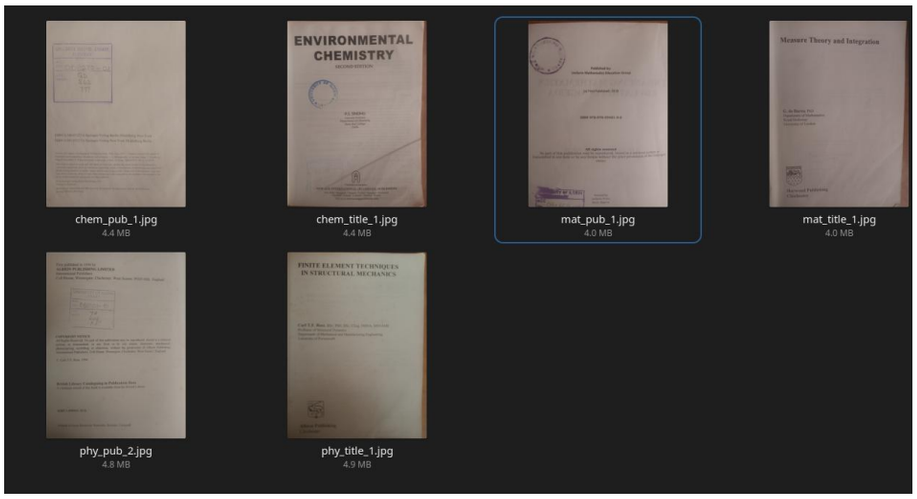


Figure 2 Samples of datasets collected

2. Binarisation: The application of binarisation techniques, particularly adaptive thresholding, proved highly effective in separating text from background. This process resulted in:
   a. Improved contrast between text and background.
   b. Enhanced text clarity, especially for books with non-uniform background colours.
   c. Reduced file sizes by an average of 65%, optimising storage and processing requirements.
3. Noise removal: Employing noise removal algorithms, such as median filtering and morphological operations, yielded the following improvements:
   a. Elimination of small artefacts and speckles
   b. Smoothing of text edges
   c. Reduction in false positives during character recognition

Figures 3 and 4 depict the results of the images before and after pre-processing.

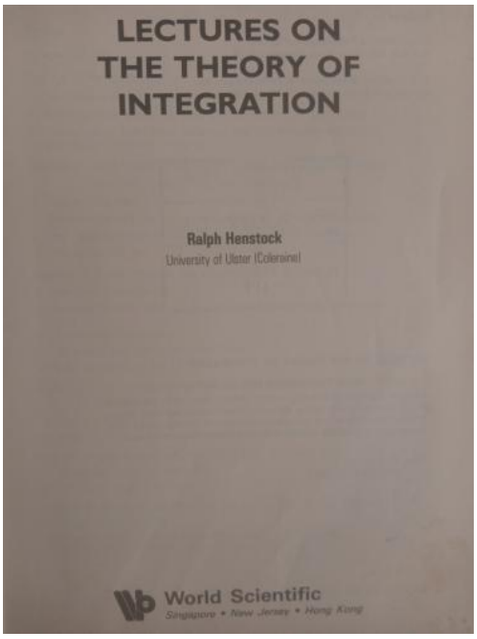


Figure 3 Image before preprocessing

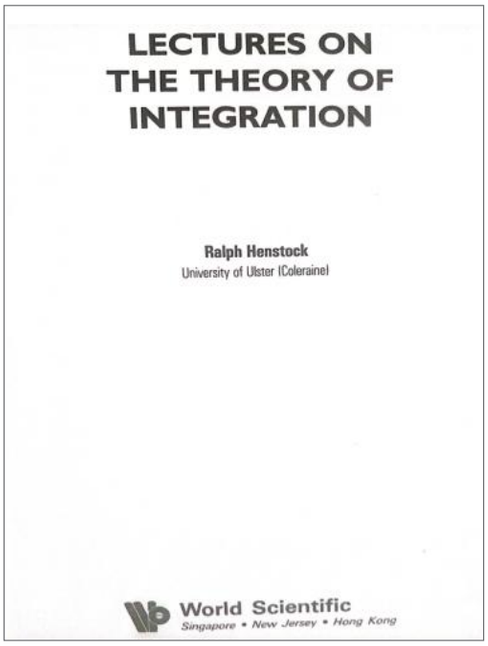


Figure 4 Image after preprocessing

The image comparison in Figures 3 and 4 visually demonstrates the effectiveness of the pre-processing techniques. The processed image shows clearer text, reduced background noise, and enhanced overall readability.

OCR Results

Tesseract OCR, an open-source OCR engine developed by Google, was used to extract text. The implementation of Tesseract in the system proved to be straightforward and effective, largely due to the robust pre-processing steps already implemented.

Tesseract OCR Implementation

Tesseract OCR was integrated into our Python-based back-end using the pytesseract library, which provides a convenient Python wrapper for the Tesseract engine. The basic implementation involved the following steps:

1. Loading the pre-processed image
2. Applying Tesseract OCR to extract text
3. Post-processing the extracted text for further analysis

Figure 5 shows the code snippet demonstrating the OCR process:

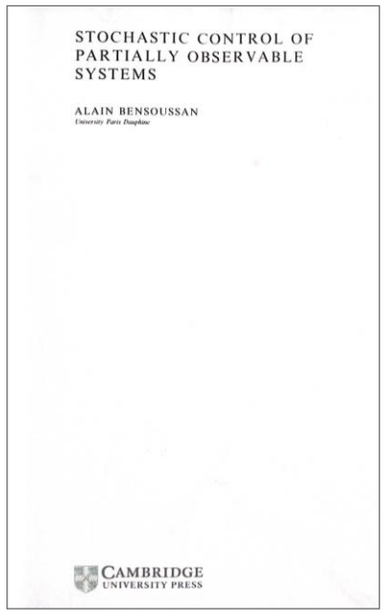
```python
def perform_ocr(image_path):
    # Load the preprocessed image
    image = Image.open(image_path)

    # Perform OCR
    text = pytesseract.image_to_string(image)

    # Basic post-processing
    text = text.strip()

    return text
```

Figure 5 Code snippet for the implementation of the OCR

After the implementation, the result from the input and output images from the OCR are depicted in Figure 6.

NER Results

To implement the Named Entity Recognition (NER) component of the system, spaCy "en_core_web_lg" model was utilised. This model was further trained to




```
(ll_env) sodiq@sodiq:~/Documents/catalog-system$ /home/sodiq/Documents/c
STOCHASTIC CONTROL OF
PARTIALLY OBSERVABLE
SYSTEMS

ALAIN BENSOUSSAN

University Paris Dauphine

8 CAMBRIDGE

oP UNIVERSITY PRESS
```

Figure 6 OCR sample input and output

specifically identify the following entities from book title and publication pages:

1. PERSON (Author)
2. International Standard Book Number (ISBN)
3. TITLE (Book Title)
4. ORG (Publisher)
5. YEAR (Publication Date)

Model Training

The spaCy model was fine-tuned on the dataset, significantly improving its performance in identifying book-specific entities. Figure 7 shows the output from training the model.

Data Cleaning and Pre-processing

After the NER process, the extracted entities required cleaning and validation to ensure data quality and consistency. An algorithm was implemented to clean the NER entities and return only valid entries, followed

```
[2021-05-07 10:56:17,101] [INFO] Created vocabulary
[2021-05-07 10:56:17,101] [INFO] Finished initializing nlp object
[2021-05-07 10:56:18,937] [INFO] Initialized pipeline components: ['tok2vec', 'ner']
✔ Initialized pipeline

============================= Training pipeline =============================
ℹ Pipeline: ['tok2vec', 'ner']
ℹ Initial learn rate: 0.001
E    #       LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
---  ------  ------------  --------  ------  ------  ------  ------
  0       0          0.00     48.83    0.00    0.00    0.00    0.00
  0     200         61.60   1450.88   87.31   94.10   81.43    0.87
  0     400        106.64    229.86   94.53   98.28   91.05    0.95
  1     600         82.88     97.06   95.41   98.65   92.37    0.95
  1     800         74.53     98.82   95.56   97.65   93.57    0.96
  2    1000         43.85     46.05   95.70   97.72   93.77    0.96
  2    1200         70.21     44.15   94.14   96.84   91.58    0.94
  3    1400         44.34     30.40   96.21   98.14   94.36    0.96
  4    1600         52.70     31.08   96.26   97.94   94.63    0.96
  5    1800         47.27     28.40   95.95   97.60   94.36    0.96
  6    2000         30.78     12.80   96.45   98.48   94.50    0.96
  8    2200         32.32     15.10   96.39   98.08   94.76    0.96
 10    2400         60.31     21.93   96.25   98.21   94.36    0.96
```

Figure 7 Output from the training model

by mapping these entities to a standardised dictionary format. Finally, the pymarc library was utilised to create and save MARC records based on the cleaned and formatted data.

Entity Cleaning and Validation Algorithm

This algorithm ensures that:

1. Multiple authors are correctly handled and joined
2. ISBNs are validated and cleaned of non-digit characters
3. Titles and organisation names are cleaned of extra spaces
4. Years are validated to be within a reasonable range (1400-2023)

Entity Mapping

After cleaning and validation, the entities were mapped to a standardised dictionary format as implemented in Figure 8.

```
142
143  mapped_entities = {
144      "author": cleaned_entities.get("PERSON", ""),
145      "isbn": cleaned_entities.get("ISBN", ""),
146      "title": cleaned_entities.get("TITLE", ""),
147      "publisher": cleaned_entities.get("ORG", ""),
148      "publish_date": cleaned_entities.get("YEAR", "")
149  }
150
```

Figure 8 Code snippet of entity mapping

This mapping ensures a consistent structure for all processed books, facilitating easier integration with the MARC record creation process.

MARC Record Creation and Storage

Using the pymarc library, MARC records from the cleaned and mapped entities will be created. The process involved:

1. Initialising a new MARC record
2. Adding fields corresponding to each entity type
3. Saving the record in MARC format
4. The MARC output is represented in Figure 9.

```
(ll_env) sodiq@sodiq:~/Documents/catalog-system$ /home/sodiq/Documents/catalog-system/ll_env/bin/pytho
=LDR  00406nam a22001453a 4500
=008  240626s2015\\\\xxu\\\\\\\\\\\000\0\eng\d
=020  \\$a01951534480
=245  04$aElements of Chemistry
=100  1\$aPenny Reid
=264  \1$bCaped Publishing$c2015
=250  \\$aeBook Second Edition
=300  \\$a1 online resource
=336  \\$atext$btxt$2rdacontent
=337  \\$acomputer$bc$2rdamedia
=338  \\$aonline resource$bcr$2rdacarrier
```

Figure 9 MARC record output

The data cleaning and formatting stage, coupled with MARC record creation, significantly enhanced the quality and usability of the extracted bibliographic data. By implementing cleaning algorithms and leveraging the pymarc library, the raw NER output was standardised and made into library-ready MARC records. This process forms a crucial bridge between the OCR and NER stages and the final cataloguing output, ensuring high-quality and consistent bibliographic records.

To assess the effectiveness and accuracy of the system developed, this study employed a range of performance evaluation metrics for OCR and NER.

Table 2 Performance metrics for OCR implementation

Area	CER	WER
Overall	3.2%	5.7%
Title pages	2.8%	4.9%
Publication pages	3.6%	6.5%

As presented in Table 2, the title pages have the lowest error rate of 2.8% for CER and 4.9% for WER, while Publication pages have 3.6% for the CER and 6.5% for WER. The overall error for the two entities is 3.2% for CER and 5.7% for WER.

Table 3 Performance metrics for NER implementation

Entity Type	Precision	Recall	F1-Score
PERSON	0.94	0.92	0.93
ISBN	0.98	0.97	0.97
TITLE	0.96	0.95	0.95
ORG	0.91	0.89	0.90
YEAR	0.99	0.98	0.98

As shown in Table 3, the precision for the year as an entity has the highest percentage of 99%, followed by ISBN with 98%, while TITLE has 96%, PERSON has 94%, and ORG has the least precision of 91%. From the results, it can be deduced that the system was able to detect all the entities accurately, with YEAR having the best accuracy.

The performance of the proposed system was further compared with existing works to show the accuracy and effectiveness of the system developed. Three different works were used for the basis of our comparison, and the performance metrics were compared. Table 4 gives a comparative analysis with other works.

System Deployment

The front end of our automated book cataloguing system was developed using React, a popular JavaScript library for building user interfaces and styled with TailwindCSS, a utility-first CSS framework. This combination allowed for the rapid development of a responsive and visually appealing interface while maintaining a high degree of customisation. Some of the key features are Image Upload Functionality, MARC Record Review and Edit Interface. Download and Update Options

Image Upload Component

The image upload component allows users to upload two crucial pages from a book:

1. The title page
2. The publication information page

A sample of this is depicted in Figure 10.

MARC Record Review and Edit Interface

Once the back-end processing is complete, the frontend receives a JSON representation of the MARC record. This data is then presented in an editable interface, allowing librarians to review and modify the extracted information if necessary.

1. Key features of this interface include:
2. Editable fields for each MARC data element
3. Validation to ensure data integrity

Table 4 Comparative analysis of the developed model with existing works

Study	Method	Precision	Recall	F1-Score	CER	WER
This Study	OCR+NER+EnhancedImageProcessing	0.960	0.940	0.950	3.2	5.7
Tian et al. [18]	Transformer-based NER	-	-	0.890	-	-
Carbune et al. [20]	OCR+ entity recognition+ XLM-RoBERTa	-	-	0.880	-	-
Alghemei et al. [15]	OCR+YOLOv8	0.912	-	-	-	-

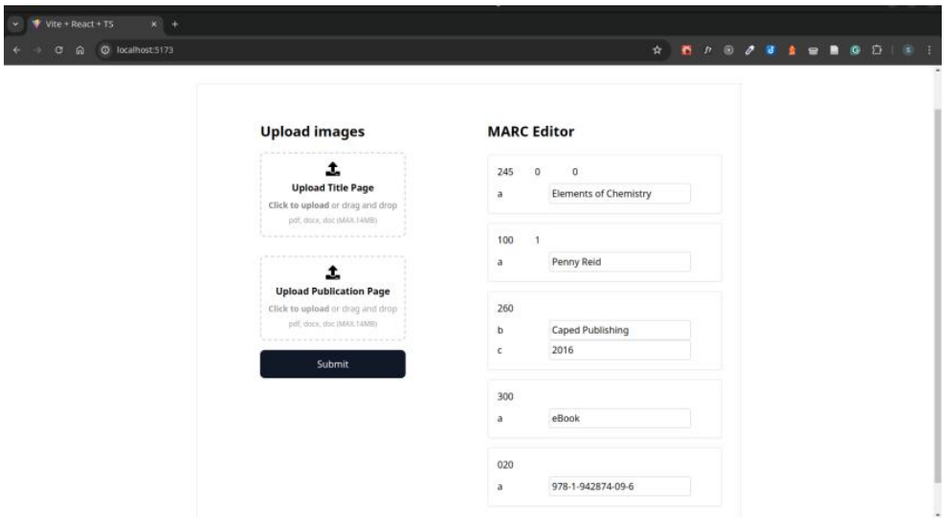


Figure 10 Image upload Interface

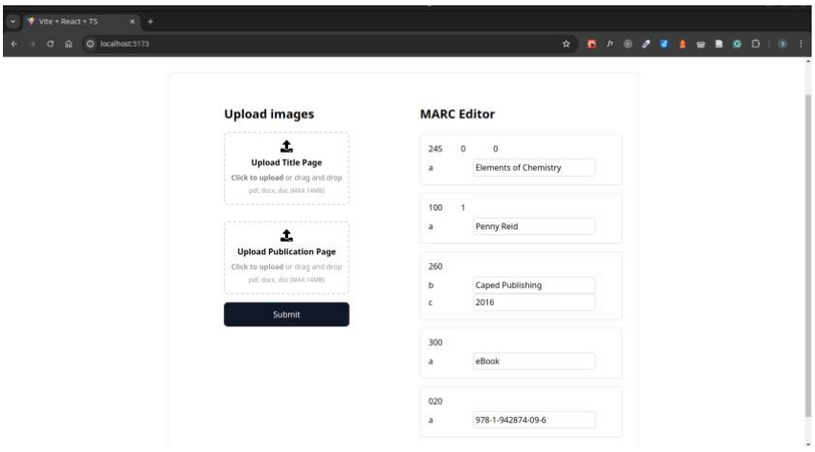


Figure 11 MARC record review/edit interface

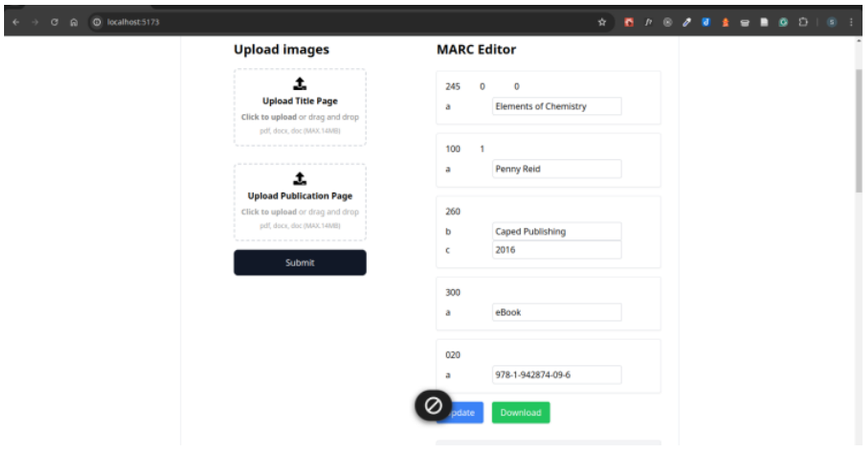


Figure 12 Download and update options for processed MARC records

Download and Update Options

After reviewing and editing any necessary columns, users can choose to:

1. Download the MARC record as a file or
2. Update the record in the system

Figure 12 depicts the interface for downloading and updating MARC records.

The front-end development, leveraging React and TailwindCSS, resulted in a user-friendly, efficient, and visually appealing interface for the automated book cataloguing system. The combination of image upload and a flexible MARC record editing interface streamlines the cataloguing process.

CONCLUSION

By utilising techniques in image processing and NLP techniques, this study demonstrates the potential to reduce the time and effort required for book cataloguing while maintaining high levels of accuracy. The successful deployment of this system with a user-friendly interface and standardised MARC record generation positions this study as a valuable tool for modern libraries. While the current implementation shows promising results, there is room for further enhancement and expansion. The recommendations outline pathways for improving the system's versatility, accuracy, and integration capabilities. As libraries evolve in the digital age, automated solutions like the one presented in this study will be crucial in maintaining efficient, accurate, and up-to-date catalogues. In conclusion, this automated book cataloguing system represents a step forward in library automation, offering a balance of efficiency and accuracy that can benefit library operations. As the system continues to evolve and adapt to diverse library needs, it has the potential to transform the cataloguing process, allowing librarians to focus more on user services and collective development, ultimately enhancing the overall library experience for both staff and patrons.

REFERENCES

[1] S. Haider, "Library cataloguing, classification, and metadata research: a bibliography of doctoral dissertations," *Cataloguing & Classification Quarterly,* vol. 58, no. 1, pp. 28-43, 2020.

[2] S. S. Gundakanal and M. Kaddipujar, "August and December " *Technology,* vol. 3, no. 2&3, 2019.

[3] D. E. Orbih and A. J. Aina, "Issues, benefits and challenges of original cataloguing versus copy cataloguing: The experience at the Lagos State University," *International Journal of Library and Information Science,* vol. 6, no. 5, pp. 88-97, 2014.

[4] H. Mamman *et al.*, "Beyond the Seeds: Fairness Testing via Counterfactual Analysis of Non-Seed Instances," *IEEE Access,* 2024.

[5] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Pearson, 2016.

[6] S.B. Junaid, A.A. Imam, A.O. Balogun, ...N.A.I. Kakumi, and S. Mahamad, "Recent Advancements in Emerging Technologies for Healthcare Management Systems: A Survey," Healthcare (Basel), vol 3, no. 10, 2022, doi: 10.3390/healthcare10101940.

[7] S.B. Junaid, A.B. Imam, S. Basri, ...N. Kakumi, and A. Alazzawi, "Artificial Intelligence, Sensors and Vital Health Signs: A Review," Applied Sciences, vol. 12, 2022, doi: 10.3390/app122211475.

[8] S.B. Junaid, A.A. Imam, A. Balogun, ...N. Kakumi, and A. Hashim, "Recent Advances in Artificial Intelligence and Wearable Sensors in Healthcare Delivery", Applied Sciences, vol. 12, 2022, doi: 10.3390/app122010271.

[9] K. Sharifani and M. Amini, "Machine learning and deep learning: A review of methods and applications," *World Information Technology and Engineering Journal,* vol. 10, no. 07, pp. 3897-3904, 2023.

[10] A. A. Adejo and A. Y. Misau, "Application of artificial intelligence in academic libraries in Nigeria," 2021.

[11] S. Barsha and S. A. Munshi, "Implementing artificial intelligence in library services: a review of current prospects and challenges of developing countries," *Library Hi Tech News,* vol. 41, no. 1, pp. 7-10, 2023.

[12] A. Tella, O. Akanmu Odunola, and L. WO, "Cataloguing and classification in the era of artificial intelligence: Benefits, and challenges from the perspective of cataloguing librarians in Oyo State, Nigeria," *Vjesnik bibliotekara Hrvatske,* vol. 66, no. 1, pp. 159-176, 2023.

[13] M. Humbel, J. Nyhan, A. Vlachidis, K. Sloan, and A. Ortolja-Baird, "Named-entity recognition for early modern textual documents: a review of capabilities and challenges with strategies for the future," *Journal of Documentation,* vol. 77, no. 6, pp. 1223-1247, 2021.

[14] M. Short, "Text Mining and Subject Analysis for Fiction; or, Using Classification, Keyword Extraction, and Named Entity Recognition to Assign Subject Headings to Dime Novels," 2019.

[15] N. Alghemei, H. Alkhadafe, and I. Nasir, "Efficient Real-time Book Cover Text Extraction with YOLOv8 and SpaCy," in *2024 IEEE 4th International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)*, 2024: IEEE, pp. 768-772.

[16] S. Faizullah, M. S. Ayub, S. Hussain, and M. A. Khan, "A survey of OCR in Arabic language: applications, techniques, and challenges," *Applied Sciences,* vol. 13, no. 7, p. 4584, 2023.

[17] B. Kiessling, G. Kurin, M. T. Miller, and K. Smail, "Advances and Limitations in Open Source Arabic-Script OCR: A Case Study," *arXiv preprint arXiv:2402.10943,* 2024.

[18] S. Tian, A. Erdengasileng, X. Yang, Y. Guo, Y. Wu, J. Zhang, J. Bian, and Z. He, "Transformer-based named entity recognition for parsing clinical trial eligibility criteria", in Proceedings of the 12th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics (BCB '21), New York, NY, USA, pp. 1–6, 2021, doi: https://doi.org/10.1145/3459930.3469560

[19] A. Llabrés, A. U. Dey, D. Karatzas, and E. Valveny, "Image-text matching for large-scale book collections," in *International Workshop on Document Analysis Systems*, 2024: Springer, pp. 89-102.

[20] V. Carbune *et al.*, "Chart-based reasoning: Transferring capabilities from llms to vlms," *arXiv preprint arXiv:2403.12596,* 2024.

[21] S. Lafia, D. A. Bleckley, and J. T. Alexander, "Digitising and parsing semi-structured historical administrative documents from the GI Bill mortgage guarantee program," *Journal of documentation,* vol. 79, no. 7, pp. 225-239, 2023.