International Journal of Innovative Research in Computer Science and Technology (IJIRCST)
ISSN (Online): 2347-5552, Volume-14, Issue-1, January 2026
DOI: https://doi.org/10.55524/ijircst.2026.14.1.2
Article ID IRP-1699, Pages 9-16
www.ijircst.org

Handwritten Notes Digitization and Multilingual Summarization Using OCR and Generative AI

Suchetha N V[1], Anisha Upadhayaya H S[2], Anushree U Rao[3], H N Swati[4], and Mallana Gowda G S[5]

[1]Associate Professor, Department of Computer Science & Engineering, Sri Dharmasthala Manjunatheshwara Institute of Technology, Ujire, Karnataka, India

[2,3,4,5] BE Scholar, Department of Computer Science & Engineering, Sri Dharmasthala Manjunatheshwara Institute of Technology, Ujire, Karnataka, India

Correspondence should be addressed to Suchetha N V; itsmesuchethanv@gmail.com

Received: 19 November 2025 Revised: 7 December 2025 Accepted: 22 December 2025

ABSTRACT- This research proposes a fully client-side system for digitising and summarising handwritten notes that integrate optical character recognition (OCR) with advanced natural language generation models. Initial experimentation with a Convolutional Neural Network (CNN)– based handwritten character recognition system revealed significant challenges, including low recognition accuracy, strong dependence on dataset quality, and poor generalisation across diverse handwriting styles. A subsequent implementation using Gemini 2.5 Pro provided high-quality English summaries but failed to deliver equivalent performance for multilingual academic content. To overcome these issues, this study adopts a hybrid methodology combining Tesseract.js OCR for multilingual text extraction with Gemini 2.5 Flash for fast, context-aware, and language-flexible summarization. The system is able to take handwritten notes, convert them into clear and readable text, and produce well-structured academic summaries across different subjects and languages. Our tests show that it works reliably, adapts well to multiple languages, and is easy to use, making it a strong and practical tool for educational digitization.

KEYWORDS- Handwritten Note Digitization, OCR, CNN, NLP, Multilingual Processing, Summarization, Deep Learning.

I. INTRODUCTION

Personal handwritten documents often provide students with tangible copies of their personal notes, classroom discussions, concepts, formulas, and diagrams, which they can reference in the future for projects and exam studying. Even with the rise of technology, students often select the written method given that it promotes topic retention and understanding. That said, handwritten notes can be difficult in that they are often unorganized, and do not provide concise, editable, or searchable information. Because of this, the digitization of handwritten notes into an organized structure has been a focal point of research, especially with the ever-growing implementation of AI in education.

In the early stages of this project, handwritten text digitization was approached using traditional OCR tools and deep learning-based Handwritten Text Recognition (HTR). The first model implemented was a CNN-based handwritten character recognizer, chosen because CNNs are well-known for their strong performance in pattern recognition tasks. During the study, it was observed that actual handwritten notes are not as simple as clean and properly arranged datasets. Since the CNN model was trained on a limited set of handwriting samples, it struggled with common real-world issues such as noisy backgrounds, inconsistent writing styles, cursive scripts, different pen or pencil types, and even shadows or uneven lighting in captured images.. Because of these issues, the output was often inconsistent, and the CNN approach was not reliable unless a much larger dataset was collected and the model was trained again extensively. To overcome these challenges, the research shifted toward using Tesseract.js—a browser-based OCR engine powered by LSTM networks, which are better suited for sequence-based text recognition. Tesseract provided a practical balance of accuracy and speed, handled varied handwriting styles more effectively, and supported multiple languages without requiring custom training. With text extraction handled more reliably, the next focus was generating clear, meaningful summaries from the extracted content, which often appeared incomplete or unstructured.

At first, advanced language models like Gemini 2.5 Pro were used for summarization. While the model produced coherent and accurate summaries, its strong performance was limited mostly to English. This became a major concern because many students take notes in regional languages such as Kannada or
Hindi, or even in a mix of multiple languages. To address this, Gemini 2.5 Flash was adopted instead, offering multilingual
summarization, faster response times, and a more suitable architecture for running directly in the browser.

The final system brings together Tesseract.js for OCR and Gemini 2.5 Flash for summarization, so you get a full pipeline that can digitize handwritten notes in different languages. Everything runs right in your browser. That means your notes stay private—nothing gets sent to a server. Thanks to these updates, the tool feels way more accessible and efficient, especially for students and teachers who need something reliable in a classroom setting.

II. RELATED WORK

Mehul Gupta and Kabir Soeny[1] present methods for rapid digitalization of printed and handwritten prescriptions using a combination of Google Vision API, the C-Cube Algorithm, and a 3-Step Filtering Algorithm. Their findings indicate that the C-Cube model significantly enhances performance with nearly 90% higher F-score and is approximately 588 times faster, while the 3-Step model achieves an 8,600% improvement in F-score with a 231-fold speed increase, demonstrating remarkable efficiency in medical prescription digitization.

Chen ShanWei et al. [2] proposed a CNN-based handwritten numeral recognition model trained on the MNIST dataset to identify digits used in basic arithmetic operations. The study aims to automate the grading of handwritten mathematical homework, and results show that the CNN-based system performs more accurately and reliably compared to existing grading technologies.

Constantin Lehenmeier et al. [3] focus on the digitization of historical handwritten weather tables using layout detection and deep learning-based OCR. Their system is capable of recognizing handwritten text with approximately 82% accuracy and detecting tabular layouts with an accuracy of 87%, thereby enabling more efficient preservation and processing of historical meteorological records.

Parashuram Bannigidad and Chandrashekar Gudada [4] describe a system for digitizing historical Kannada handwritten manuscripts. The approach uses text line segmentation, Local Binary Patterns (LBP), and classifiers such as LDA, KNN, and SVM. Their results show promising recognition accuracy, contributing significantly to the preservation and digital archiving of ancient Kannada cultural documents.

K. Amulya et al. [5] present a survey on methods used to digitize handwritten Kannada notes, employing CNN models along with image preprocessing tools like OpenCV and NumPy. Their experiments on the Chars74K dataset reveal that the CNN model achieves around 87% validation accuracy, outperforming Tesseract OCR, which achieves 86%, thereby underscoring the potential of CNNs in regional handwriting recognition.

R. P. Ram Kumar et al. [6] describe an automated handwritten text recognition system that integrates CNN, RNN, and LSTM architectures using the Keras and TensorFlow frameworks. Supported by OpenCV and OCR preprocessing, their system achieves high accuracy depending on dataset complexity, demonstrating the effectiveness of hybrid deep learning models for general handwritten text digitization.

S. Rakesh et al. [7] provide a comprehensive survey of deep-learning-based handwriting recognition methods, covering CNN, RNN, BLSTM, HMM, CRNN, and CTC approaches. Their review highlights accuracy rates ranging from 80% to 98%, outlining the technological advancements, strengths, and limitations of modern deep learning architectures in handwriting recognition tasks.

Hexel [8] presents a system for evaluating handwritten answer sheets using a combination of CNN, RNN/Transformer models, and Tesseract OCR integrated with NLP techniques. The study reports strong recognition accuracy across varied handwriting styles, making it useful for automated grading and evaluation systems in academic environments.

Habibi Dehkordi et al.[9] explore techniques to enhance summarization accuracy in medical discharge notes by adding highlight-based context into GPT-4o prompts. Their results indicate that summaries generated with highlights achieve 96% completeness—around 8% higher—and show fewer clinical errors, demonstrating the effectiveness of prompt engineering in medical NLP applications.

Abhinandan Chiney et al.[10] propose an anchor-based Multi-Channel CNN (MCCNN) model for digitizing handwritten forms. Their system achieves high accuracy rates of 93% for letters, 96% for digits, and 93% for mixed handwritten information, showcasing the effectiveness of anchor-based extraction in structured form digitization.

Loitongbam Gyanendro Singh [11] and Stuart E. Middleton introduce TrOCR-ctx, a tabular context-aware OCR model integrated with ByT5 for digitizing old tabular documents. The system achieves a Word Error Rate (WER) of 0.049 and a Character Error Rate (CER) of 0.035, which represents a drastic improvement compared to traditional OCR, particularly in reconstructing semantically aligned tabular data.

Nasuha Iskandar et al.[12] present a method for digitizing handwritten mathematical expressions using projection-based segmentation and CNN-based classification. Their system records an accuracy of 80.73% on online datasets and about 77% on offline datasets, indicating its effectiveness in converting mathematical notations into structured digital expressions.

Josua Käser et al. [13] looked into how well different models handle multilingual healthcare text summarization. They tested GPT-3, Multilingual BERT, and Biomedical T5. Out of all of them, GPT-3 stood out. It delivered the clearest, most useful summaries, and it didn't even need a translation step beforehand. That makes it a strong pick for working with clinical documents in different languages.

Maria Clara Saad Menezes et al.[14] analyze the ability of GPT-4 to interpret medical notes in English, Spanish, and Italian hospital settings. The model achieves a 79% agreement rate with clinical experts, with the highest accuracy in Spanish (88%), followed by Italian (84%), and English (77%), showing GPT-4's potential for multilingual clinical text analysis.

Supriyono et al. [15] took a deep dive into 233 text-summarization studies, using the PRISMA approach to guide their survey. They looked at everything from old-school rule-based techniques to machine learning and the latest transformer models. Along the way, they called out some big challenges—things like semantic drift, adapting to specific domains, and making sense of information from multiple documents at once. They also pointed out where the field's headed, laying out fresh directions for researchers who want to push summarization quality further.

III. PROBLEM STATEMENT

Writing stuff down by hand still matters - whether you're in class, at work, or studying alone. Even though putting pen to paper helps your brain soak up info and remember it longer, those scribbles can cause headaches once you need to go back through them. Messy lettering, cramped spacing, gaps in meaning, or faded ink make old pages hard to dig into again - or pass along. When notebooks pile up, finding what actually counts turns into a wild goose chase, wasting time and zapping mental energy right before tests.

On top of that, going through scribbled notes by hand takes forever - people end up skimming piles of disorganized info with no help spotting key ideas. Current apps fail to offer one smooth system that turns messy writing into clean digital text while also pulling out useful summaries for fast review. Optical character readers often choke on sloppy script, joined-up letters, mixed languages, or odd styles, leaving behind broken or wrong transcriptions. When the converted words can't be trusted, students struggle to use smart learning helpers or keep reliable backups of their class notes.

Because of these limits, we really need a smart tool that turns messy handwriting into neat digital words while making short, clear summaries without losing key ideas. Instead it uses OCR plus powerful language tools like Google Gemini to handle rough hand-written stuff and help students turn scattered notes into tidy, usable formats fast. This kind of setup tackles the main issue - poor handling of paper notes - and boosts access, speed, and how well people learn today at school.

IV. OBJECTIVES OF THE PROPOSED SYSTEM

The objectives below are written to match your system (OCR + Gemini Summarization):

- To accurately extract text from handwritten notes using a browser-based OCR engine capable of handling diverse handwriting styles and varying image qualities.
- To generate clear, coherent, and academic-quality summaries using an advanced language model designed to interpret raw OCR output effectively.
- To support multilingual summary generation so that students can receive outputs in their preferred language for better understanding and accessibility.
- To simplify the study process by converting lengthy handwritten pages into concise digital summaries that enhance revision efficiency.
- To provide a complete client-side solution ensuring user privacy, fast processing, and no dependency on external server infrastructure.
- To improve the management and organization of notes by enabling users to store, download, and revisit processed content easily.
- To reduce the time and effort required for manual digitization, helping students focus more on learning rather than rewriting or summarizing content.

V. METHODOLOGY

The methodology of the proposed system is structured into five major stages: image acquisition, preprocessing, OCR-based text extraction, cleaning of extracted text, and AI-driven summarization. The workflow is designed to ensure that handwritten information is captured accurately, processed robustly, and transformed into high-quality summaries suitable for academic use.

The next step uses OCR to pull out text, handled by the Tesseract.js tool. It checks the image you upload, spots hand-written letters, then turns them into digital text. We go with Tesseract because it works well across languages and adapts to different ways people write. Still, the output can come out messy - uneven gaps, broken words, or extra marks - thanks to sloppy writing or blurry images.

To tackle these issues, the system uses a solid prep step for text. It clears out clutter and odd characters while adjusting line layouts. Spacing mistakes get fixed at this point. Broken or half-read words are rebuilt right here. Good cleanup makes the pulled text clearer and better organized. That way, the summary tool gets clean and legible data every time.

After cleaning, the text gets passed to Gemini's summary tool using a backend link. This smart system takes plain input and turns it into smooth, clear paragraphs that keep meaning intact. Instead of just English, it works in multiple languages like Kannada and Hindi too. Because of this, more people can use it no matter their main language. On top of that, summaries focus on specific topics so students get key ideas fast - without reading piles of hand-written notes.

After making the summary, the app shows it on screen right away. Then again, it saves both original text and condensed version into a log - so you can come back later if needed. On top of that, having past entries saved helps keep notes tidy over time.

The last step is testing the system on different handwriting types, school topics, notes in multiple languages, also images taken at odd angles, with bad light or shadows. So it works well even when conditions aren't perfect. By using this layered approach, the tool turns messy hand-written pages into clean digital files that fit today's classrooms better. It connects analog note-taking with smart tech without slowing things down.

VI. SYSTEM DESIGN

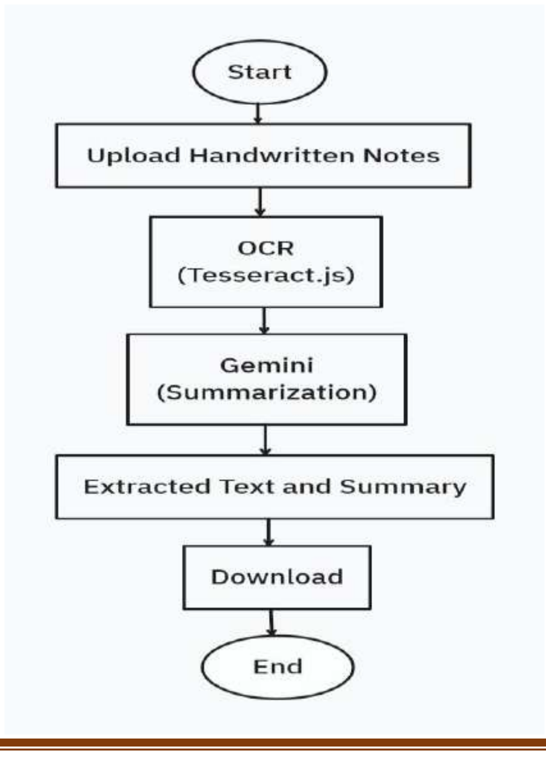

Figure 1: system architecture of the Handwritten Notes

In the above figure 1, it shows the Illustrates the system architecture of the handwritten notes digitization and summarization spplication. The handwritten notes scanner and summary tool works from start to finish in one smooth flow, letting people turn paper scribbles into clean digital text along with smart auto-summaries. Instead of separate parts, it combines user interface actions, handwriting recognition, artificial intelligence condensing, saving files, plus delivering results - all linked together step by step. Everything's built around being simple to operate, getting details right, while moving info quickly behind the scenes.

The action starts on the front screen, once someone signs in to reach their main panel. After logging in, they might choose to add a scanned note instead of checking older digests saved inside. Choosing upload pulls up a tool letting them pick pictures with hand-drawn letters. That picture moves next into a conversion chain handling written text.

After upload, the picture goes into an OCR part built with Tesseract.js. This step matters - it connects messy handwriting to digital words. Instead of skipping ahead, the tool checks each mark, picks out letters, then pulls plain text from the file. Next, that text gets sent off for shortening. When reading fails or grabs almost nothing, the app warns the person and asks for a better photo.

Once the data is pulled out, it heads to the Gemini AI via the backend API for summarizing. That part takes messy text and turns it into clear, logical write-ups. It grabs main points, groups them smoothly, then builds clean paragraphs using smart structuring. Instead of just dumping info, it shapes it into something easy to follow and school-friendly. Users can pick their language - like English, Kannada, or Hindi - for the final version. Because of this flexibility, lots of learners across regions can actually use it.

After sorting things out, the cleaned-up text plus brief overview head back to the front-end screen part. The interface shows everything in a straightforward way so people can go over it or save it later. There's a button that lets you grab the end result as a file, just in case you want it on your device without needing the web.

At the same time, it stores the pulled text along with its summary in MongoDB - creating a database you can search later. That way, people can go back to earlier summaries instead of uploading files again or processing them from scratch. When someone uses the "View History" button on the dashboard, the history feature pulls up those stored results. When folks are done checking or saving what they found, they just tap logout to leave safely.

The setup keeps things moving without hiccups - log in, send files, grab text, boil it down, show results, save them. One piece works on its own, so boosting scan precision doesn't mess with smart summary updates. Tweak storage space? Do it later, no problem. This whole layout handles scrawled notes fast, turns them into clean digital bits automatically. It just runs steady, grows when needed, feels natural to use.

A. Algorithms Used and Pipeline Logic

The handwritten note conversion system relies on three main algorithms: an OCR tool that pulls out text, followed by a pre-processing step to clean it up - then finally a generative AI that creates organized summaries. Each part runs one after another, building a simple but effective flow designed specifically to work well inside web browsers.

- OCR (Optical Character Recognition) – Tesseract.js: The initial phase converts handwritten input into digital format through Tesseract.js. Being an open-source tool, this OCR engine identifies both printed and hand-drawn letters. Instead of treating the whole image at once, it splits the scanned note into sections for individual symbols. Each section gets compared to trained models based on LSTM networks - producing matched text plus accuracy ratings. While supporting various languages, it also adapts to different penmanship styles; hence fitting well with student-made material. Such extracted data then becomes the base for later steps like analysis or condensing information.
- Text Preprocessing Algorithm: Following OCR processing, extracted text may include errors like messy gaps, split words, extra symbols - caused by poor handwriting or low-quality images. Because of this, a special cleanup method is used. The tool strips out irrelevant marks and distortions, adjusts inconsistent spaces and returns between lines, joins pieces of cut-off words, while fixing mismatches from shaky lettering. Unreadable signs are dropped, plus layout consistency is enforced across the document. Turning rough OCR results into neat, organized content helps clarity; it boosts precision and value of summaries made afterward.
- AI Summarization – Gemini 2.5 Flash Model: After preprocessing, the text goes to the Gemini 2.5 Flash Model; this system uses advanced algorithms to summarize effectively. Instead of just shortening text, it detects main themes while mapping how points connect logically. By filtering out repetition, it builds smooth summaries in structured paragraphs. While focusing on clarity, it works across several languages - like English, Kannada, or Hindi - for broader accessibility. As a result, users receive brief yet complete overviews suited for fast review and better understanding.

VII. SYSTEM IMPLEMENTTION SNAPSHOTS

This section includes several visual snapshots of the implemented Smart Note Scan application, illustrating the user experience and final results.

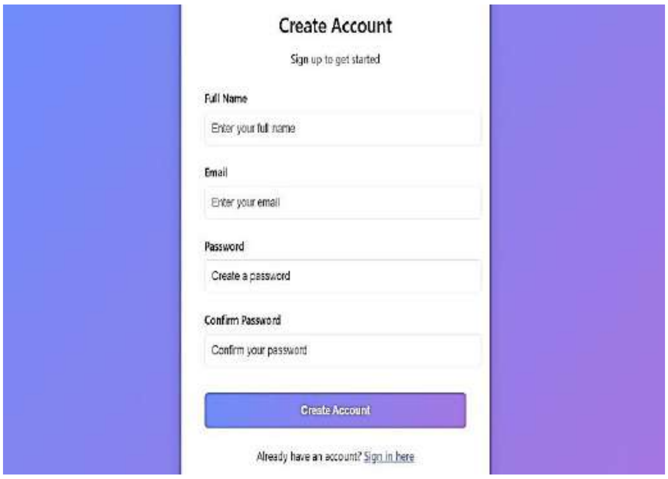

Figure 2: User Registration Interface

Figure 2 illustrates the User Registration page of the

SmartNoteScan system. This interface enables new users to create an account by entering their full name, email address, password, and confirmation password. The registration module ensures secure and personalized access to the application. By validating user credentials at the time of account creation, the system associates all uploaded handwritten notes, extracted text, summaries, and history records with the respective user. This module forms the entry point to the system and ensures controlled access to system resources.

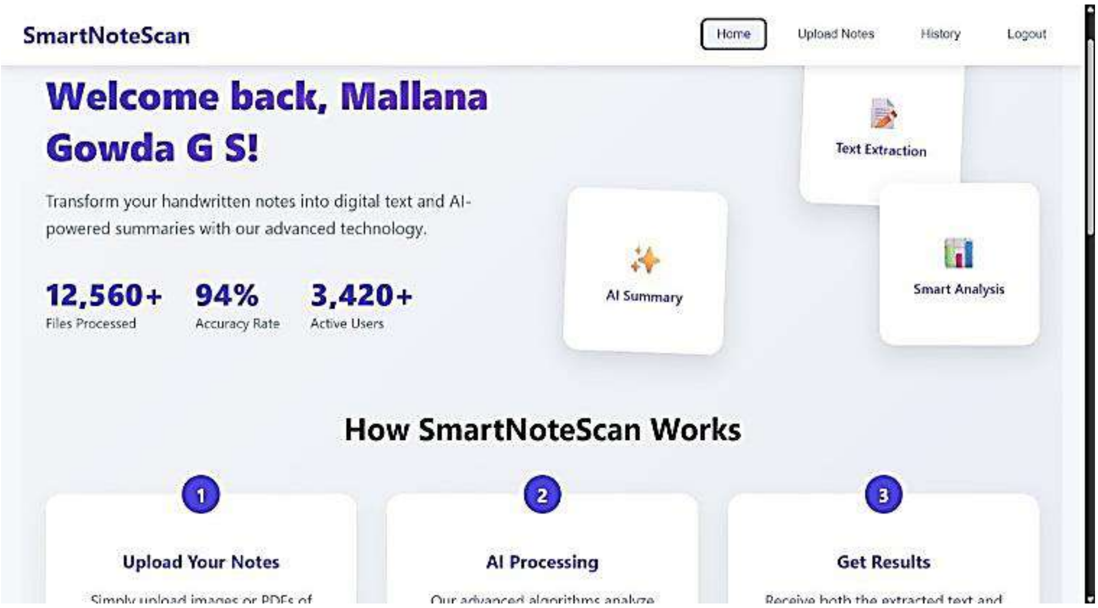

Figure 3: Home Dashboard Interface

Figure 3 shows the Home Dashboard displayed after successful user login. This page welcomes the user and provides an overview of the system's functionality. The dashboard highlights core services such as Text Extraction, AI Summary, and Smart Analysis. It also displays system statistics like number of files processed, accuracy rate, and active users, reflecting system reliability. The "How SmartNoteScan Works" section visually explains the three-step workflow: uploading handwritten notes, AI processing, and obtaining results. This dashboard acts as the central navigation hub of the application.

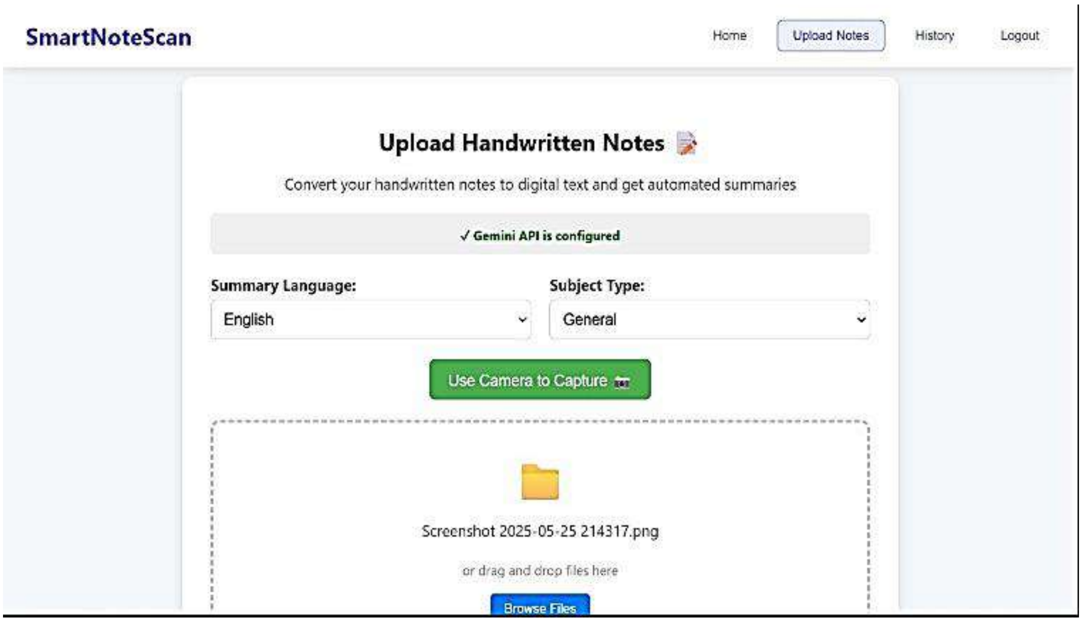

Figure 4: Upload Handwritten Notes Interface

Figure 4 represents the Upload Handwritten Notes interface. This screen allows users to upload handwritten notes strictly in image format. Users can select the summary language and subject type before processing. The interface also provides an option to capture images using the device camera. Once an image is uploaded, it is forwarded to the OCR module for processing. This interface bridges user input with the backend OCR and summarization pipeline.

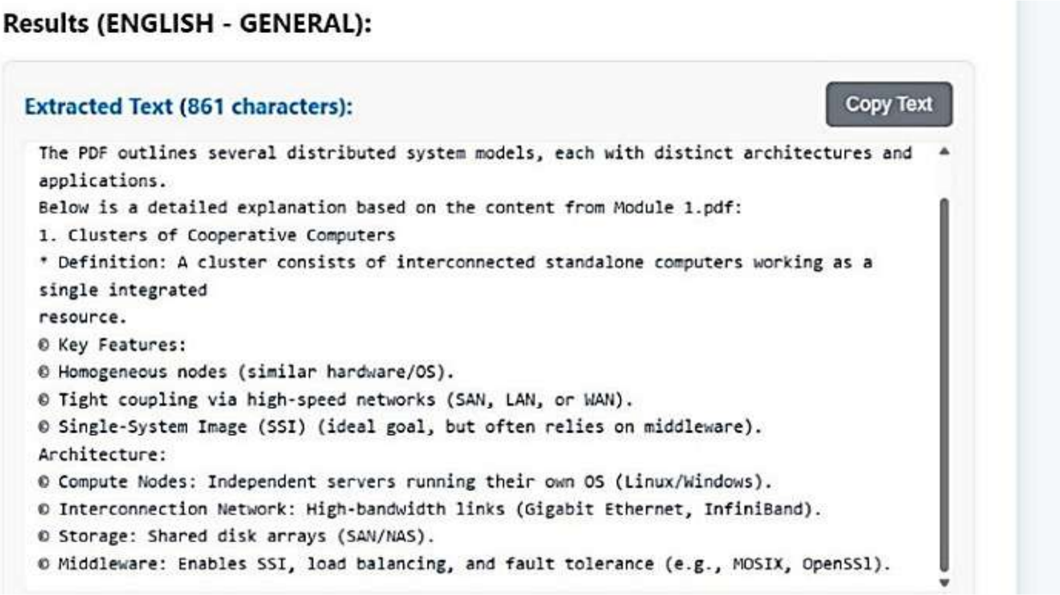

Figure 5: Extracted Text Output Interface

Figure 5 displays the Extracted Text generated by the OCR module using Tesseract.js. The extracted text is shown along with the total character count, allowing users to verify the OCR results. A "Copy Text" option is provided for convenience. This stage ensures transparency in the digitization process by allowing users to view the raw digital text obtained from handwritten images before summarization.

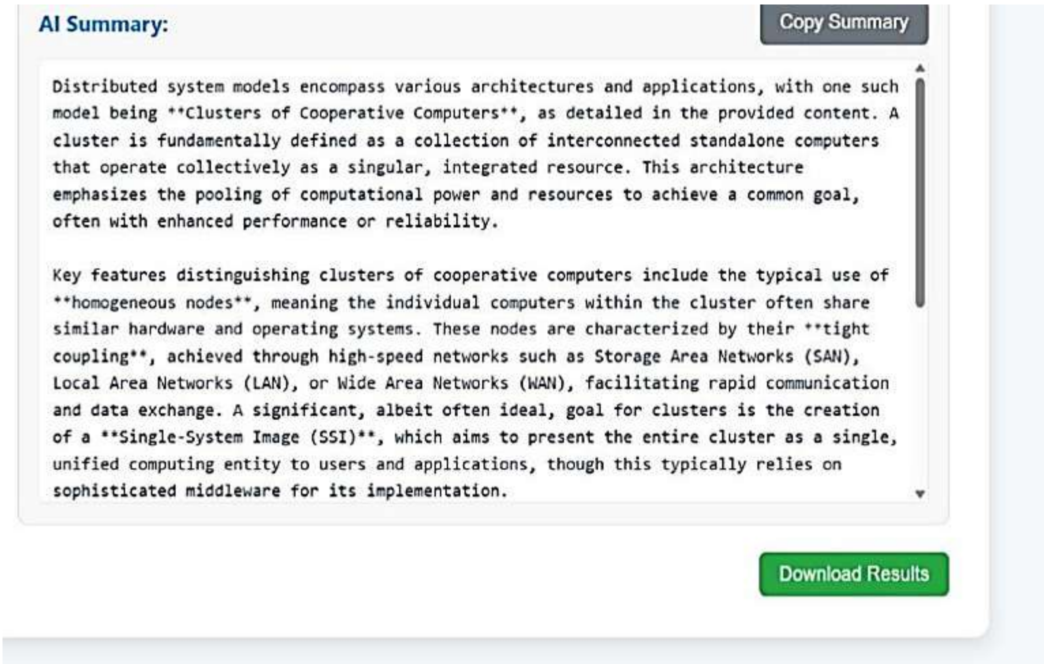

Figure 6: AI Summary Output Interface

Figure 6 presents the AI Summary generated by the Gemini AI model. The summarization module converts the cleaned OCR text into well-structured, continuous paragraphs suitable for academic study. The summary highlights key concepts and explanations derived from the handwritten notes. Features such as "Copy Summary" and "Download Results" allow users to easily store and reuse the summarized content. This interface demonstrates the effectiveness of AI in transforming lengthy handwritten material into concise, readable summaries.

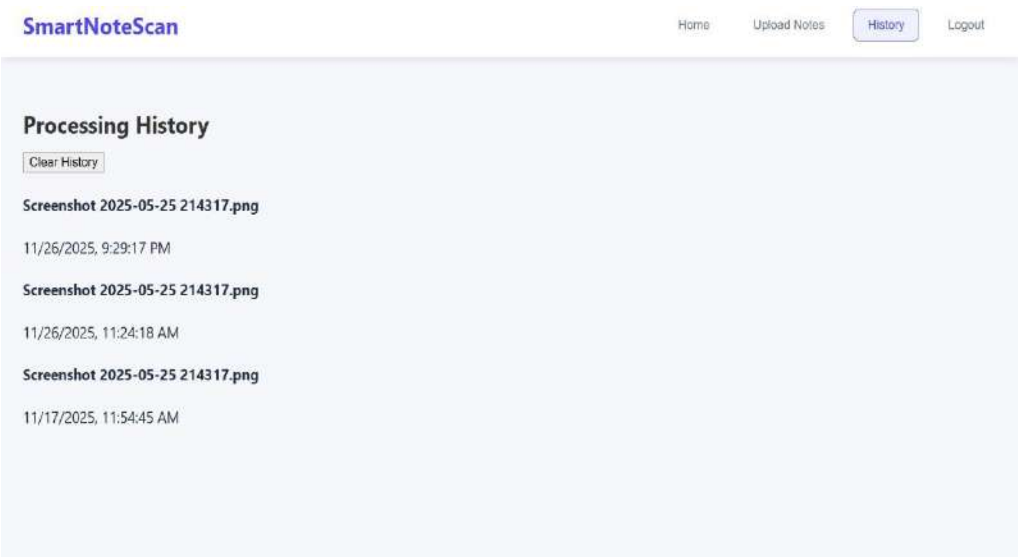

Figure 7: Processing History Interface

In the above Figure 7 shows the Processing History page of the system. This module maintains a log of previously uploaded handwritten images along with their processing timestamps. Users can revisit earlier results without re-uploading files, improving efficiency and usability. The "Clear History" option allows users to manage stored records. This interface demonstrates effective storage and retrieval of user-specific processing data.

VIII. CONCLUSIONS

This research presents a complete client-side solution for the digitization and summarization of handwritten academic notes. Through extensive testing, this study clearly shows the shortcomings of CNN-based handwritten recognition, especially when dealing with the wide variety of real-world handwriting and the absence of large, high-quality training datasets. Shifting to Tesseract.js greatly improved the consistency of text extraction while also providing multilingual capabilities and a simpler deployment process. For summarization, Gemini 2.5 Pro initially delivered strong results, but its limited language support made it unsuitable for a diverse user base. Replacing it with Gemini 2.5 Flash offered a more practical and balanced solution, with support for multiple languages, faster processing, and reliable summarization quality.

By integrating Tesseract.js for OCR with Gemini 2.5 Flash for summarization, the project delivers a strong, scalable system that can benefit students, teachers, and educational institutions. Since the entire workflow runs on the client side, the system protects user privacy, works across different platforms, and can be used instantly without any server setup. Overall, this solution successfully connects handwritten notes to digital learning tools, making study processes more efficient and supporting the advancement of modern educational technology. Future enhancements may include incorporating domain-specific language models, improving OCR accuracy through custom handwriting datasets, and extending the system to support diagrams, mathematical expressions, and multimodal inputs. The findings of this study confirm that integrating OCR with generative AI provides a powerful solution for transforming handwritten notes into structured digital knowledge.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] M. Gupta and K. Soeny, "Algorithms for rapid digitalization of prescriptions," *Visual Informatics*, vol. 5, no. 3, 2021. Available from: https://doi.org/10.1016/j.visinf.2021.07.002

[2] S. L. Chen ShanWei and D. A. Ng Theam Foo, "A CNN based handwritten numeral recognition model for four arithmetic operations," *Procedia Computer Science*, vol. 192, 2021. Available from: https://doi.org/10.1016/j.procs.2021.09.218

[3] C. Lehenmeier, M. Burghardt, and B. Mischka, "Layout detection and table recognition—Recent challenges in digitizing historical documents and handwritten tabular data," *Lecture Notes in Computer Science*, 2020. Available from: https://doi.org/10.1007/978-3-030-54956-5_17

[4] P. Bannigidad and C. Gudada, "Digitization and recognition of historical Kannada handwritten manuscripts using text line segmentation with LBP features," *Journal of Emerging Technologies and Innovative Research*, vol. 6, no. 4, 2019. Available from: https://tinyurl.com/mrydhfym

[5] K. Amulya *et al.*, "A survey on digitization of handwritten notes in Kannada," *International Journal of Innovative Technology and Exploring Engineering*, vol. 12, no. 1, 2022. Available from: https://tinyurl.com/37fujfe4

[6] R. P. Ram Kumar *et al.*, "Automated handwritten text recognition," *E3S Web of Conferences*, vol. 430, 2023. Available from: https://doi.org/10.1051/e3sconf/202343001022

[7] S. Rakesh *et al.*, "Handwritten text recognition using deep learning techniques: A survey," *MATEC Web of Conferences*, vol. 392, 2024. Available from: https://doi.org/10.1051/matecconf/202439201126

[8] P. Bala, "Handwritten character recognition and evaluation," *Journal of Artificial Intelligence Practice*, vol. 4, 2025.

[9] M. K. Habibi Dehkordi *et al.*, "Improving large language models' summarization accuracy by adding highlights to discharge notes: Comparative evaluation," *JMIR Medical Informatics*, vol. 13, 2025. Available from: https://medinform.jmir.org/2025/1/e66476

[10] A. Chiney *et al.*, "Handwritten data digitization using an anchor-based multi-channel CNN (MCCNN) trained on hybrid dataset (h-EH)," *Procedia Computer Science*, vol. 189, 2021. Available from: https://doi.org/10.1016/j.procs.2021.05.095

[11] L. G. Singh and S. E. Middleton, "Tabular context-aware optical character recognition and tabular data reconstruction for historical records," *International Journal on Document Analysis*

and Recognition, vol. 28, no. 3, 2025. Available from: https://doi.org/10.1007/s10032-025-00543-9

[12] N. Iskandar, W. J. Chew, and S. K. Phang, "The application of image processing for conversion of handwritten mathematical expression," *Journal of Physics: Conference Series*, vol. 2523, 2023. Available from: https://doi.org/10.1088/1742-6596/2523/1/012014

[13] J. Käser *et al.*, "Multilingual text summarization in healthcare using pre-trained transformer-based language models," *Computers, Materials and Continua*, vol. 83, no. 1, 2025. Available from: https://doi.org/10.32604/cmc.2025.061527

[14] M. C. S. Menezes *et al.*, "The potential of generative pre-trained transformer-4 (GPT-4) to analyse medical notes in three different languages: A retrospective model-evaluation study," *The Lancet Digital Health*, vol. 7, no. 1, 2025. Available from: https://doi.org/10.1016/S2589-7500(24)00246-2

[15] A. Supriyono *et al.*, "A survey of text summarization: Techniques, evaluation and challenges," *Natural Language Processing Journal*, vol. 7, 2024. Available from: https://doi.org/10.1016/j.nlp.2024.100070

ABOUT THE AUTHORS

Dr. Suchetha N V has completed her B.E in the year 2011, MTech in the year 2015 from VTU Belagavi, and Ph.D. in the year 2024 from VTU Belagavi. Her areas of interest are Image Processing and Deep Learning. She has published 13 international research papers. She is the life member of Kannada Sahitya Parishad.

Anisha Upadhayaya H S received her B.E. degree in Computer Science and Engineering from Visvesvaraya Technological University (VTU), Belagavi. Her areas of interest include Deep Learning, the Internet of Things (IoT), and Artificial Intelligence (AI). She was a student in the Department of Computer Science and Engineering at SDM Institute of Technology, Ujire.

Anushree U Rao received her B.E. degree in Computer Science and Engineering from Visvesvaraya Technological University (VTU), Belagavi. Her areas of interest include Deep Learning, the Internet of Things (IoT), and Artificial Intelligence (AI). She was a student in the Department of Computer Science and Engineering at SDM Institute of Technology, Ujire.

H N Swati received her B.E. degree in Computer Science and Engineering from Visvesvaraya Technological University (VTU), Belagavi. Her areas of interest include Deep Learning, the Internet of Things (IoT), and Artificial Intelligence (AI). She was a student in the Department of Computer Science and Engineering at SDM Institute of Technology, Ujire.

Mallana Gowda G S received his B.E. degree in Computer Science and Engineering from Visvesvaraya Technological University (VTU), Belagavi. His areas of interest include Deep Learning, the Internet of Things (IoT), and Artificial Intelligence (AI). He was a student in the Department of Computer Science and Engineering at SDM Institute of Technology, Ujire.