IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Cognitive Image Processing For Bank Passbook And Form Automation

[1]B. Jaya Deepika, [2]G. Jahnavi, [3]G. Yuva Teja Sree, [4]A. V. D. Harini, [5]Dr. L. Ganesh

[1]Under Graduate Student, [2]Under Graduate Student, [3]Under Graduate Student, [4]Under Graduate Student, [5]Associate Professor,

[1]Electronics and Communication Engineering,

[1]Gayatri Vidya Parishad College of Engineering for Women, Visakhapatnam, India.

Abstract: Cognitive Image Processing (CIP) analyzes and extracts meaningful data from images, aiding sectors like banking and healthcare. The manual entry of bank details into forms is prone to typographical errors and is time consuming. This paper presents an automated approach using CIP to extract relevant details from bank passbook images and auto fills bank forms. The system integrates image processing techniques to enhance text clarity and Optical Character Recognition (OCR) to extract relevant details with Natural Language Processing (NLP) techniques, including Regular Expressions (Regex) and Levenshtein Distance to enhance text extraction accuracy. The extracted data is used to automatically fill the forms. A user-friendly mobile application is developed using Replit. The application auto-fills bank forms, reducing typographical errors and improving efficiency. Users can capture passbook images, verify extracted details, and edit as needed. This solution streamlines form-filling, minimizing errors and saving time.

Index Terms **– Cognitive Image Processing, Tesseract OCR, NLP, Regex, Levenshtein Distance, Bank Form Automation.**

1. INTRODUCTION

Banking processes often require customers to manually enter details from their passbooks into withdrawal and deposit forms. This process is often time consuming and prone to typographical errors. Cognitive Image Processing (CIP) aims to automate this by extracting relevant information from bank passbook images and using it to auto fill banking forms such as withdrawal and deposit forms. The proposed system leverages image processing, Tesseract OCR for text extraction and NLP techniques [5] such as Regex and Levenshtein Distance for text refinement and correction. The system enhances efficiency and accuracy in data entry, reducing human intervention and potential errors. There is no direct reference for the implementation of this work so, to develop a reliable and effective solution, specific techniques such as Tesseract OCR [1], Regular Expressions (Regex) [3], BERT [2] and Levenshtein Distance [4] are adapted from existing research papers. These individual techniques, drawn from the literature, are integrated into the custom workflow to achieve accurate and automated data extraction from passbook images.

2. IMPLEMENTATION METHODOLOGY

This system is implemented in two phases. In the first phase, passbook samples of two banks are collected. Image processing techniques are applied on these samples to enhance text clarity. Two OCR models, Tesseract OCR and Easy OCR are used to extract text from preprocessed images. Since the extracted text often contain errors and unwanted characters, NLP techniques are employed to clean the extracted text. Specific fields like Name, Account Number, Branch Code and IFSC Code are extracted using Regex and Spacy. To correct errors in IFSC Code and extract accurate branch names, Levenshtein Distance and BERT are used.

In the second phase, a mobile application is developed using Replit, with UI design prototyped in Figma. The application allows users to capture or upload images of their bank passbook and automatically extracts and fills relevant data into pre-designed bank forms. The form templates for withdrawal and deposit slips are created using HTML and CSS, while JavaScript is used to dynamically insert the extracted data into the form fields. This system provides an efficient, user-friendly solution to reduce manual entry errors and streamline bank form-filling processes.

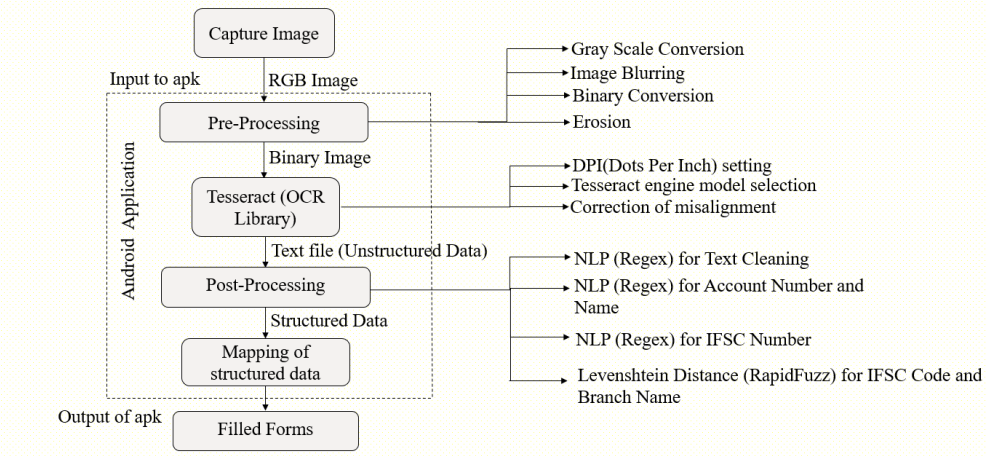


Fig. 2.1 Flow Chart of Implementation

2.1 IMAGE PREPROCESSING

Image processing is a technique used to analyze, enhance, manipulate and extract information from digital images. The primary goal of image processing is to improve image quality, identify patterns and transform visual data into structured information that can be further analyzed. Images of bank passbooks undergo preprocessing techniques such as grayscale conversion, noise reduction, thresholding and post morphological operations to enhance text clarity and improve OCR accuracy.

2.2 TEXT EXTRACTION

Optical Character Recognition (OCR) is a technology used to extract text from images, scanned documents, and handwritten notes. The enhanced images were then subjected to OCR to extract textual data. There are many OCR models, for this work two models Tesseract OCR and Easy OCR are used.

2.3 TEXT CLEANING WITH NLP

Text cleaning is a crucial preprocessing step in NLP that ensures raw text is structured and free from unnecessary noise. OCR introduces errors such as real word and non-word errors. The extracted text is cleaned using NLP techniques to remove special characters, unnecessary symbols and noise.

2.4 RELEVANT TEXT EXTRACTION

Relevant text extraction is a fundamental task in form filling that involves identifying and retrieving key pieces of information from unstructured text. Regular Expressions (Regex) and Spacy are used to extract specific fields such as Name, Account Number, Branch Code and IFSC Code.

2.5 ERROR CORRECTION IN EXTRACTED TEXT

Text extracted from sources such as OCR often contains errors due to various factors like poor image quality, font distortions and OCR misinterpretations. A matching algorithm using Levenshtein distance and RapidFuzz library is applied to correct errors in extracted IFSC codes and to extract relevant branch name.

2.6 APPLICATION DEVELOPMENT

A mobile application is created using Replit. The application allows users to upload or capture images of passbooks, extract relevant data and preview auto-filled bank forms. Form templates for withdrawal and deposit forms are designed using HTML and CSS. Extracted data is dynamically inserted into form fields using JavaScript.

3. RESULTS

The system was tested using multiple passbook samples from two banks: SBI and Union Bank. The OCR results improved significantly after applying image preprocessing techniques. Comparison of Regex and Spacy for data extraction showed that Regex performed better for fields like Name, Account Number, IFSC, and Branch Code. Levenshtein distance yielded more accurate IFSC and branch name matching than BERT-based models in our use case. The final application successfully auto-filled both SBI and Union Bank forms. The system demonstrated high accuracy and reduced processing time.

3.1 IMAGE PREPROCESSING

To improve the accuracy of text extraction, various image preprocessing techniques are applied to enhance the quality of the passbook images. Grayscale conversion, binarization, noise reduction, and morphological operations were implemented to reduce noise, improve contrast, and enhance text clarity for better OCR recognition. Weighted Method (Luminance Method) is used for Gray Scale Conversion. Gaussian Filter is applied on the gray scale images of the samples. Otsu's Thresholding is employed on denoised images. Finally, Erosion with (2,2) kernel is used on the image.

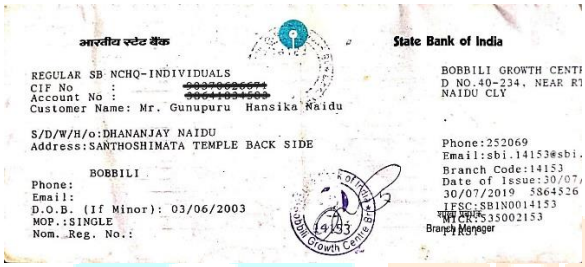


Fig 3.1.1. Sample of State Bank of India

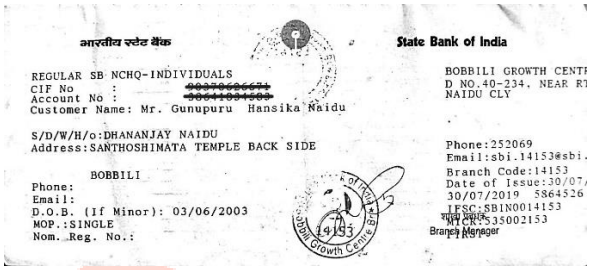


Fig 3.1.2. Gray Scale Image of Sample

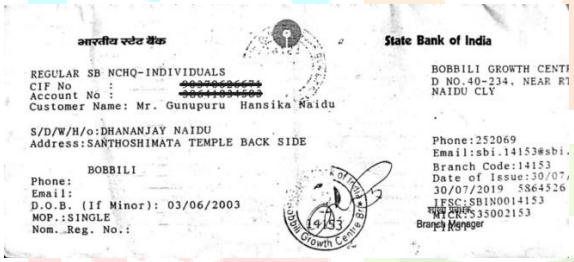


Fig 3.1.3. Denoised Image of Sample

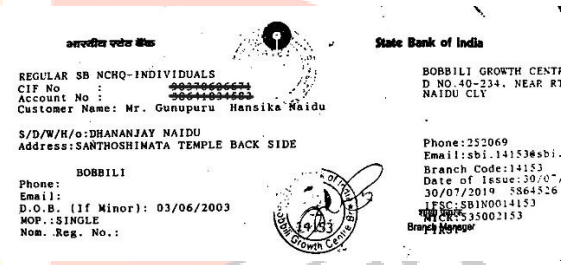


Fig 3.1.4. Binary Image of Sample

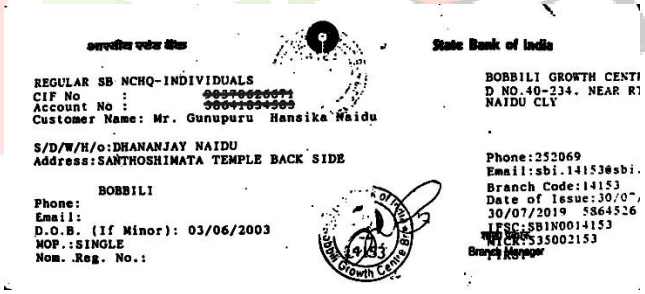


Fig 3.1.5. Eroded Image of Sample

3.2 TEXT EXTRACTION USING OCR

Optical Character Recognition (OCR) is used to extract text from preprocessed images. Tesseract OCR and Easy OCR are two widely used OCR engines. Tesseract OCR performance depends on image quality and it is useful on simple documents and printed text. Easy OCR is good at recognizing text from handwritten text. Both OCR models are used on the samples.

Fig 3.2.1. OCR Output of Sample using Tesseract OCR and Easy OCR

To determine which OCR model between Tesseract OCR and Easy OCR performs best for extracting text from bank passbooks, their accuracy is evaluated using a defined mathematical equation. Both OCR models were applied to the pre-processed passbook images and measured their performance in terms of character recognition accuracy. Tesseract OCR model delivers higher accuracy. This evaluation helped in selecting the most efficient OCR model for automating bank form filling with minimal errors. The predefined mathematical equation to find accuracy is mentioned below:

$$ACCURACY = \frac{NO.\ OF\ LETTERS\ CORRECTLY\ EXTRACTED}{TOTAL\ NO.\ OF\ LETTERS} * 100$$

Table 3.2.1. Accuracy Comparison of Tesseract OCR and Easy OCR Models

BANK	TESSERACT OCR	EASY OCR
STATE BANK OF INDIA	95.26%	78.76%
UNION BANK OF INDIA	92.46%	70.32%

3.2 TEXT CLEANING USING NLP

OCR introduces errors such as real word and non-word errors. To improve the accuracy of extracted text from bank passbooks, Natural Language Processing (NLP) techniques is used to clean the OCR output. The raw text obtained from OCR often contains unwanted symbols, special characters, and noise, which can affect further data extraction processes.

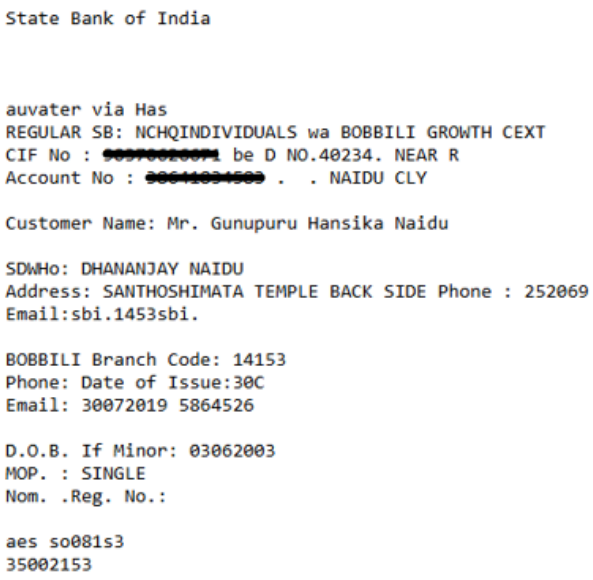


Fig 3.3. Cleaned Text using NLP

3.4 RELEVANT TEXT EXTRACTION

Regular Expressions (Regex) and Spacy are employed to extract relevant information such as name, account number, branch code, and IFSC code from bank passbooks After comparing the performance of both methods, Regex proved to be more effective in accurately identifying structured information like account numbers, branch codes, and IFSC codes. The accuracy of both techniques was evaluated using the same predefined mathematical equation used for OCR model evaluation.

To ensure accurate IFSC code extraction, Levenshtein Distance and BERT (Bidirectional Encoder Representations from Transformers) were used. Two datasets were created for SBI and Union Bank branches in Vizag, Vizianagaram, and Srikakulam to correct and extract the IFSC Code and Branch name. Levenshtein Distance outperformed BERT in terms of accuracy, making it the preferred method for extracting correct IFSC codes and branch names in the automated bank form-filling process.

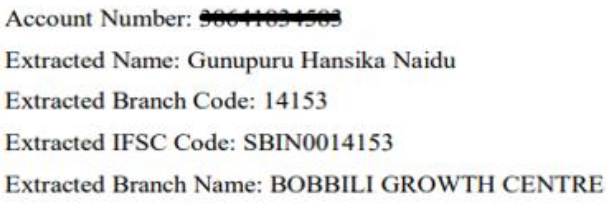


Fig 3.4.1. Extracted Relevant Text

Table 3.4.1. Accuracy Comparison of Regex and Spacy

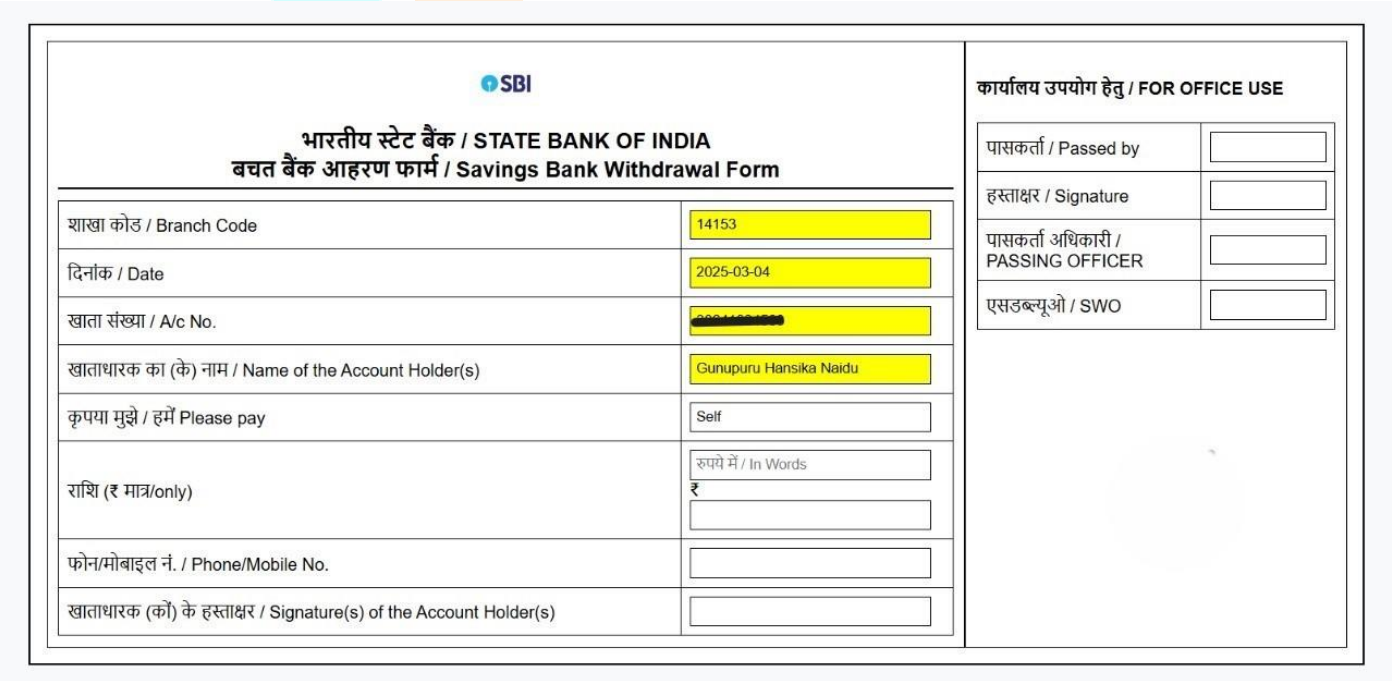
BANK	REGEX	SPACY
STATE BANK OF INDIA	98.64%	83.75%
UNION BANK OF INDIA	94.01%	79.8%

Table 3.4.2. Accuracy Comparison of Levenshtein Distance and BERT

BANK	LEVENSHTEIN DISTANCE	BERT
STATE BANK OF INDIA	100%	60.76%
UNION BANK OF INDIA	100%	62.54%

3.5 FORM CREATION AND AUTO FILLING

As part of the bank form automation process, withdrawal and deposit form templates of State Bank of India (SBI) and Union Bank of India were collected. These forms were then digitally recreated using HTML and CSS to maintain the original structure and layout. To automate the form-filling process, JavaScript is used to dynamically insert the extracted relevant information (such as name, account number, branch code, and IFSC code) into the appropriate fields of the digital forms. Additionally, to enhance visibility and user experience, the auto-filled text was highlighted, making it easy for users to verify the extracted data before submission. This automation significantly reduces manual effort and minimizes errors in bank form filling.


SBI

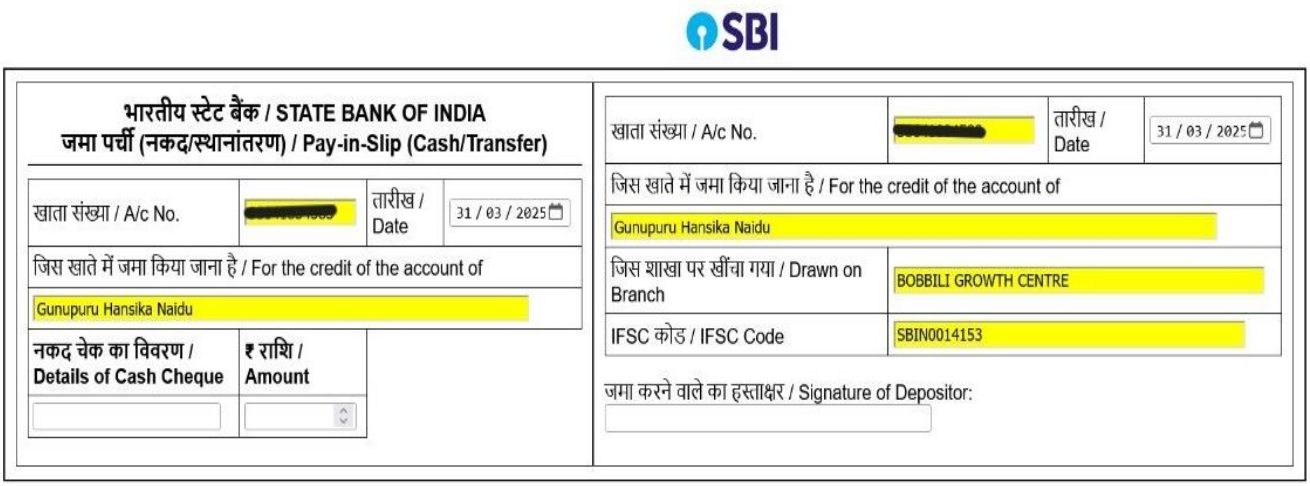
भारतीय स्टेट बैंक / STATE BANK OF INDIA
बचत बैंक आहरण फार्म / Savings Bank Withdrawal Form

शाखा कोड / Branch Code	14153
दिनांक / Date	2025-03-04
खाता संख्या / A/c No.	[redacted]
खाताधारक का (के) नाम / Name of the Account Holder(s)	Gunupuru Hansika Naidu
कृपया मुझे / हमें Please pay	Self
राशि (₹ मात्र/only)	रुपये में / In Words ₹
फोन/मोबाइल नं. / Phone/Mobile No.	
खाताधारक (कों) के हस्ताक्षर / Signature(s) of the Account Holder(s)	

कार्यालय उपयोग हेतु / FOR OFFICE USE

पासकर्ता / Passed by	
हस्ताक्षर / Signature	
पासकर्ता अधिकारी / PASSING OFFICER	
एसडब्ल्यूओ / SWO	

Fig 3.5.1. Withdrawal Form of Sample


SBI

भारतीय स्टेट बैंक / STATE BANK OF INDIA
जमा पर्ची (नकद/स्थानांतरण) / Pay-in-Slip (Cash/Transfer)

खाता संख्या / A/c No.	[redacted]	तारीख / Date	31 / 03 / 2025
जिस खाते में जमा किया जाना है / For the credit of the account of			
Gunupuru Hansika Naidu			

नकद चेक का विवरण / Details of Cash Cheque	₹ राशि / Amount

खाता संख्या / A/c No.	[redacted]	तारीख / Date	31 / 03 / 2025
जिस खाते में जमा किया जाना है / For the credit of the account of			
Gunupuru Hansika Naidu			
जिस शाखा पर खींचा गया / Drawn on Branch	BOBBILI GROWTH CENTRE		
IFSC कोड / IFSC Code	SBIN0014153		

जमा करने वाले का हस्ताक्षर / Signature of Depositor:

Fig 3.5.2. Deposit Form of Sample

3.6 APPLICATION DEVELOPMENT

The developed application successfully integrates the functionalities implemented in Phase 1, offering an end-to-end automated solution for bank form filling. The application allows users to either upload or capture images of their passbooks through a mobile interface. These images are processed in real-time to extract relevant text data such as Name, Account Number, IFSC Code, and Branch Code. The bank forms for SBI and Union Bank specifically, withdrawal and deposit forms are created using HTML and CSS. The extracted data is dynamically filled into these templates using JavaScript. The application was developed and hosted on Replit, which provided an efficient platform for rapid prototyping and testing. During testing, the app consistently demonstrated high accuracy in text extraction and auto-filling capabilities, validating the effectiveness of the integrated OCR, NLP, and form-handling modules.

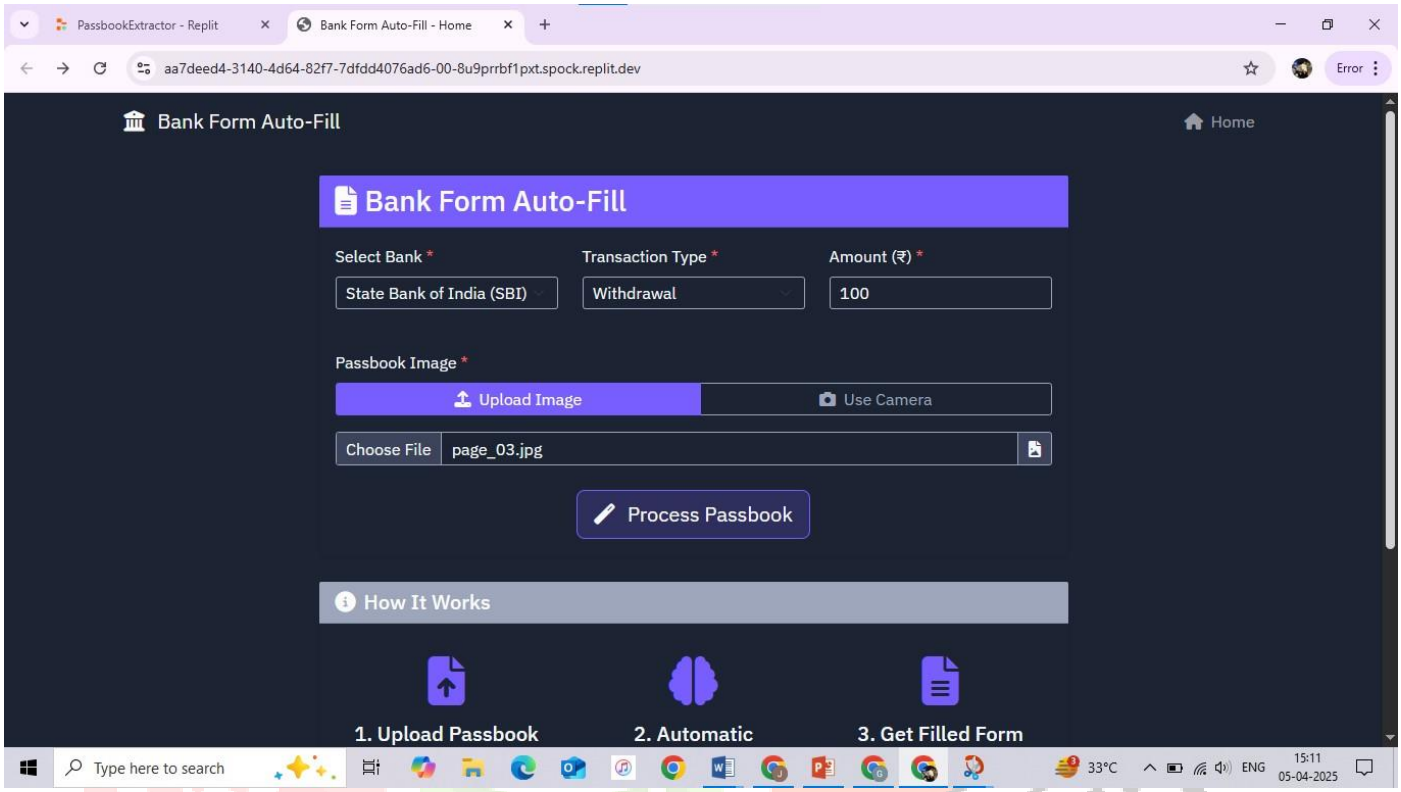


Fig. 3.6. User Interface for Bank Details Filling

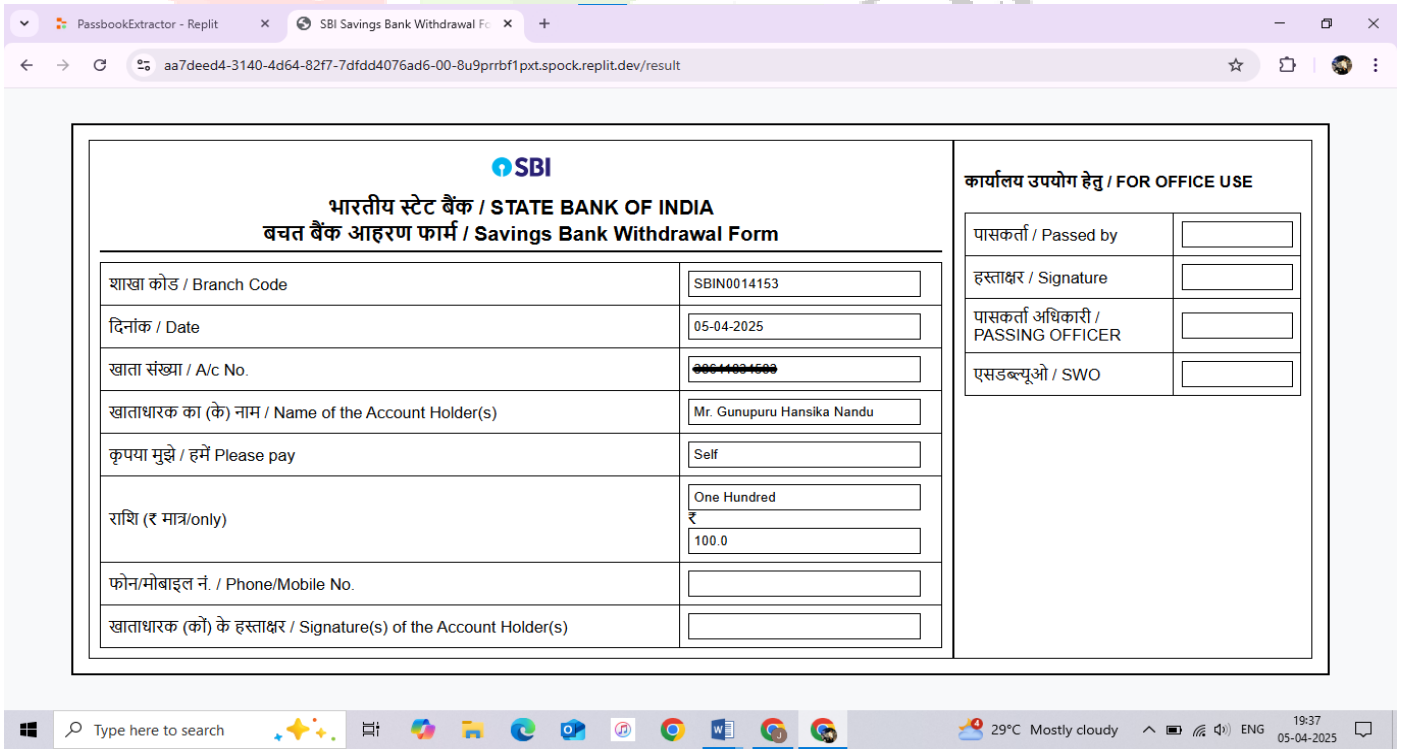


Fig 3.7. Auto-filled Form

4. CONCLUSION

This paper presented a comprehensive CIP-based approach for automating bank form filling. Samples from State Bank of India and Union Bank of India are collected and preprocessed using Gray Scale conversion, Gaussian Filter, Otsu's thresholding and erosion to improve text clarity. Two OCR models, Tesseract OCR and Easy OCR are evaluated for text extraction, with Tesseract OCR yielding better accuracy at 93.86%, compared to Easy OCR's 74.54%, making it the preferred model. For relevant text extraction, both Regex and Spacy are evaluated, with Regex achieving higher accuracy of 96.32% in extracting fields such as Name, Account Number, IFSC Code and Branch Code. Similarly, for IFSC correction and branch name extraction, Levenshtein distance achieved 100% accuracy, outperforming BERT. An application is developed with Replit, with HTML and CSS based bank form templates. Extracted information was auto-filled into the forms using JavaScript. The overall system demonstrated improved accuracy, reduced manual effort and a user-friendly interface, making it a reliable solution for automating data entry in banking workflows.

REFERENCES

[1] Image to Text Converter and Translator using Deep Learning and Image Processing, Swapnil Sinha, Harsh Kumar Kataruka, Vijayakumar Kuppusamy , 2020.

[2] Named Entity Recognition Utilized to Enhance Text Classification While Preserving Privacy, Mohammed kutbi, IEEE, 2023.

[3] A Code Error Correction System for PDF documents using Regex and similarity Matching, Lian Pan, Hao Yao, Zhipeng Li, Yuyang Ren, IEEE, 2022.

[4] Research on String Similarity Algorithm based on Levenshtein Distance, Shengnan Zhang, Yan Hu, Guangrong Bian, IEEE, 2025.

[5] A Unified Understanding of Deep NLP Models for Text Classification, Zhen Li, Xiting Wang,Weikai Yang , Jing Wu , Zhengyan Zhang , Zhiyuan Liu , Maosong Sun ,Hui Zhang , and Shixia Liu, IEEE, 2022.