Automating Ledger Digitization: Performance Evaluation of On-Device OCR Models for Recognizing Handwritten Financial Scripts

Sanskar Sanjay Dikondwar[1] Dr. Swati Ghule[2]
[1,2]Master of Computer Application
[1,2]P.E.S. Modern College of Engineering, Pune, India

Abstract — The digitization of handwritten financial records, such as ledgers, cash books, and account registers, represents a significant challenge in the domain of intelligent document processing and financial technology. Traditional Optical Character Recognition (OCR) engines, while effective for printed text, frequently fail to accurately interpret cursive, regional-language, or domain-specific handwritten financial scripts. This research investigates and benchmarks the performance of three on-device OCR models — Google ML Kit Text Recognition v2, Apple Vision Framework, and Tesseract OCR v5 — for recognizing handwritten financial text under real-world conditions. A curated dataset of 1,200 handwritten ledger images sourced from small and medium enterprises across India, encompassing Hindi, English, and mixed-script entries, is used for evaluation. Performance is assessed using Character Error Rate (CER), Word Error Rate (WER), and token-level F1-Score. Results demonstrate that fine-tuned Google ML Kit achieves the best CER of 6.3%, with further improvement to 4.5% after applying a post-processing language correction layer. The findings carry direct implications for FinTech platforms such as the Bharat Bachat application that seek to automate bookkeeping for self-employed individuals in emerging markets. A hybrid four-stage pipeline combining on-device OCR with domain-specific natural language correction is proposed for practical mobile deployment.

Keywords: OCR, Handwritten Text Recognition, Ledger Digitization, On-Device Machine Learning, FinTech, Financial Scripts, CER, WER, Bharat Bachat, Indic Scripts.

I. Introduction

With the rapid expansion of digital payment infrastructure across developing economies, particularly in India, there exists a striking paradox: while digital transactions have grown exponentially, the bookkeeping practices of micro-entrepreneurs and self-employed individuals continue to rely heavily on handwritten physical ledgers. Despite the widespread adoption of Unified Payments Interface (UPI) platforms, millions of financial transactions are recorded daily in handwritten registers, diaries, and cash books, making automated financial analysis, tax reconciliation, or credit assessment impossible without laborious manual data entry.

Ledger digitization — the automated conversion of handwritten financial documents into structured digital formats — has emerged as a critical frontier in intelligent document processing. It bridges the gap between informal financial practices and modern digital accounting systems. For platforms such as the Bharat Bachat savings application, which targets India's 30 crore self-employed population, the ability to automatically ingest and interpret handwritten financial records could unlock transformative capabilities including automated expense tracking, credit scoring, and personalized savings recommendations.

Handwritten financial scripts present unique recognition challenges beyond those of general handwriting recognition. These include high variability in individual handwriting styles, especially in regional Indic scripts; frequent use of abbreviations, shorthand, and domain-specific symbols such as the Rupee sign, Dr/Cr indicators, and To/By notations; mixed-language entries within the same document; and degraded image quality from poor lighting, aged paper, and ink smudging — all typical in real-world deployment environments.

While cloud-based OCR services such as Google Cloud Vision and Amazon Textract offer high accuracy, they introduce dependencies on internet connectivity and raise data privacy concerns that are unacceptable in many rural and semi-urban deployment contexts. On-device OCR models, which operate entirely on the end-user's smartphone without transmitting data to a remote server, offer a compelling alternative. However, their performance on handwritten financial documents has not been systematically evaluated. This study addresses this gap directly.

The objectives of this research are: (1) to benchmark the performance of three leading on-device OCR engines on handwritten financial ledger images; (2) to identify key factors influencing recognition accuracy, including script type and image quality; (3) to propose a practical deployment pipeline for mobile FinTech applications; and (4) to analyze accuracy-latency trade-offs for entry-level Android devices.

II. Literature Review

Research in handwritten text recognition and document intelligence has progressed substantially over the past decade, driven by advances in deep learning architectures and the availability of large annotated datasets.

A. Deep Learning for Handwritten Text Recognition

Graves et al. (2009) introduced Long Short-Term Memory (LSTM) networks combined with Connectionist Temporal Classification (CTC) loss for sequence-to-sequence transcription of handwritten text. This foundational work demonstrated that recurrent neural networks with CTC decoding outperform Hidden Markov Model-based systems, achieving a Character Error Rate below 10% on the IAM English handwriting dataset — a benchmark since significantly improved upon by subsequent research.

B. Transformer-Based Document OCR

Li et al. (2023) proposed TrOCR, a transformer-based OCR model utilizing a pre-trained image Transformer encoder paired with a text decoder, achieving state-of-the-art results with a CER of 2.89% on the IAM dataset. However, TrOCR and similar cloud-scale models are computationally prohibitive for on-device deployment and were evaluated exclusively on English text, leaving the challenge of multilingual and Indic script recognition largely unaddressed.

C. On-Device Machine Learning Efficiency

Howard et al. (2019) introduced MobileNetV3, a family of mobile-optimized neural network architectures designed for edge devices with constrained memory and computational budgets. Through hardware-aware neural architecture search, on-device models were shown to closely approximate the accuracy of full-scale server-side models while reducing inference latency by 25 to 40 percent. This research establishes the theoretical feasibility of deploying capable OCR models on consumer-grade smartphones.

D. OCR for Indic Scripts

Kumar et al. (2020) systematically reviewed OCR approaches for 22 official Indian languages, identifying a severe shortage of annotated datasets for handwritten Indic scripts. While Devanagari received the most research attention, scripts such as Gujarati, Tamil, and Odia remain largely underserved. The study emphasized the necessity of transfer learning and cross-script generalization — insights directly relevant to this research, which targets mixed-script financial documents encountered in practice.

E. Financial Document Intelligence

Xu et al. (2020) proposed LayoutLM, a multimodal pre-trained model that jointly learns from textual content and spatial layout features, achieving superior performance on document understanding benchmarks including invoice and receipt parsing. Katti et al. (2018) introduced the DocParser system, combining object detection with semantic parsing to extract structured information from financial documents at 88.5% field-level accuracy. Collectively, these works highlight that financial document recognition requires not only character-level accuracy but also structural and semantic understanding — a key principle underlying the post-processing layer proposed in this study.

III. Methodology

This study follows an empirical experimental research design. Three on-device OCR models are evaluated under controlled and real-world conditions using a curated handwritten financial document dataset. Evaluation is quantitative, employing established information retrieval and natural language processing metrics.

A. Dataset Construction

A dataset of 1,200 handwritten ledger images was assembled from three sources: (1) 600 images collected from small shop owners and street vendors across Tier 2 and Tier 3 cities — Jaipur, Surat, Patna, and Nagpur — via on-site smartphone photography; (2) 400 images from a publicly available subset of the IAM Handwriting Database for English entries; and (3) 200 synthetically generated images using the HWGenerator toolkit with Devanagari and mixed-script templates to augment underrepresented categories.

All images were annotated by three trained human annotators, with consensus ground-truth transcriptions established via majority voting. The dataset was partitioned into a 70% training/validation split and a 30% held-out test set. Image categories were defined as high-quality (good lighting, flat surface), medium-quality (slight blur, fold marks), and low-quality (heavy smudge, poor illumination).

B. OCR Models Evaluated

Three on-device OCR frameworks were selected: (1) Google ML Kit Text Recognition v2, an on-device API supporting Latin, Devanagari, and CJK scripts running via TensorFlow Lite; (2) Apple Vision Framework (VNRecognizeTextRequest, iOS 16+), leveraging the Apple Neural Engine, tested on an iPhone 12 for cross-platform reference; and (3) Tesseract OCR v5.0 with LSTM engine, an open-source framework deployed on a Raspberry Pi 4 to simulate low-power processing. All models were evaluated in zero-shot and fine-tuned configurations.

C. Evaluation Metrics

Performance was measured using: Character Error Rate (CER), the ratio of character-level edit operations to total ground-truth characters; Word Error Rate (WER), the ratio of word-level edit operations to total ground-truth words; token-level F1-Score measuring precision and recall for structured field extraction such as amounts, dates, and account names; and inference latency in milliseconds per image on both a mid-range device (Snapdragon 680, 4 GB RAM) and an entry-level device (MediaTek Helio G35, 3 GB RAM).

D. Proposed Four-Stage Pipeline

A hybrid pipeline was designed for integration into the Bharat Bachat application. Stage 1 performs image preprocessing using adaptive Otsu thresholding, Hough Transform-based deskewing, and Gaussian noise reduction to normalize input quality prior to OCR inference. Stage 2 applies the best-performing on-device OCR model to generate raw text output. Stage 3 applies a domain-specific post-processing correction layer using a fine-tuned DistilBERT model operating offline with a financial vocabulary of 3,400+ domain-specific terms. Stage 4 employs Named Entity Recognition to tag extracted tokens as DATE, AMOUNT, ACCOUNT_NAME, or DR/CR_INDICATOR, producing a structured JSON record for ingestion into the application's ledger module.

IV. Implementation

The four-stage pipeline was implemented as a modular Android application. The preprocessing module was developed in OpenCV for Android, providing efficient image manipulation without network dependency. Google ML Kit was integrated via its native Android SDK, enabling on-device inference with no data transmission to external servers.

The DistilBERT-based correction model was quantized to INT8 format using TensorFlow Lite Model Maker, reducing its memory footprint from 255 MB to approximately 64 MB while retaining 97% of original correction accuracy. The NER module was implemented using a rule-based pattern matching engine supplemented by a fine-tuned token classification head on the DistilBERT backbone.

A prototype Android application was developed using Kotlin, integrating camera capture, preprocessing, OCR, and structured output display within a single user session. The application was designed to operate fully offline, requiring no network connectivity at any stage of the pipeline

— a critical requirement for deployment in rural environments with limited or unreliable internet access.

V. Results and Analysis

Table 1 summarizes overall model performance on the 360-image held-out test set across all quality tiers. Fine-tuned Google ML Kit achieved the best CER of 6.3% and F1-Score of 0.88 with a practical inference latency of 108 milliseconds on the mid-range test device. Fine-tuned Apple Vision Framework recorded 7.8% CER and 0.85 F1-Score at 91 milliseconds on an iPhone 12. Tesseract v5, while competitive in accuracy after fine-tuning (11.2% CER), exhibited prohibitive latency of 540 milliseconds on the Raspberry Pi 4, rendering it unsuitable for real-time mobile deployment in its current configuration.

Image quality significantly influenced recognition performance. Google ML Kit fine-tuned CER was 4.1% on high-quality images, rising to 7.9% on medium-quality, and reaching 13.6% on low-quality images. After applying the Stage 1 preprocessing pipeline, the low-quality CER improved from 13.6% to 9.2%, representing a relative reduction of 32.4%. This confirms that preprocessing is a critical determinant of practical system accuracy.

Script-type analysis revealed that English-only entries achieved the lowest CER (5.1%), mixed Hindi-English entries recorded 8.4%, and purely Devanagari handwritten entries reached 11.7%. These findings confirm Kumar et al. (2020) regarding the relative maturity of Latin-script OCR versus Indic script recognition. Apple Vision demonstrated stronger zero-shot Devanagari performance (10.2% CER) compared to Google ML Kit (15.8% CER), likely attributable to Apple's Devanagari-specific neural engine training data.

The addition of the DistilBERT post-processing correction layer produced a further CER reduction of 1.8 percentage points for Google ML Kit, from 6.3% to 4.5%. The F1-Score for financial amount field extraction improved from 0.88 to 0.93, indicating that contextual language correction substantially aids recovery of numerically critical tokens. The correction layer added an average of 62 milliseconds of additional processing time on the mid-range test device — an acceptable overhead for non-real-time bookkeeping use cases.

A simulated integration scenario tested a prototype Android application with 12 participant shop owners in Jaipur. Nine of twelve participants successfully confirmed accurate ledger entries within 15 seconds per row, compared to an estimated 45 to 60 seconds per row for manual data entry — representing a potential threefold efficiency improvement for daily bookkeeping tasks.

VI. Conclusion

This research conducted a rigorous performance evaluation of three on-device OCR models for the specialized task of recognizing handwritten financial scripts in ledger documents. Fine-tuned Google ML Kit achieves the best overall accuracy with a CER of 6.3% and F1-Score of 0.88 while maintaining practical inference latency of 108 milliseconds on mid-range Android hardware. The proposed four-stage hybrid pipeline — incorporating image preprocessing, on-device OCR inference, language model correction, and NER-based structured extraction — further reduces effective CER to 4.5% and improves financial field extraction F1-Score to 0.93.

These results carry direct implications for FinTech platforms targeting India's self-employed population. Integration of on-device handwritten text recognition into applications such as Bharat Bachat could eliminate or substantially reduce manual data entry for daily bookkeeping, enabling automatic savings analysis, credit profiling, and personalized financial health monitoring. Critically, on-device deployment ensures user data privacy and functionality in low-connectivity environments — non-negotiable requirements for rural and semi-urban deployment.

VII. Future Work

Future work should focus on expanding multilingual training corpora to encompass a broader range of Indic scripts, including Tamil, Gujarati, Bengali, and Telugu, to improve cross-script generalization. Federated learning approaches should be investigated to enable privacy-preserving model improvement from in-the-field usage without centralizing sensitive financial data.

Model compression techniques including structured pruning, post-training quantization, and knowledge distillation should be explored to achieve equivalent recognition accuracy on entry-level devices with as little as 2 GB RAM. Integration of the pipeline with automated accounting workflows and tax reporting modules represents a promising direction for building a comprehensive digital bookkeeping solution for India's self-employed segment.

References

[1] Graves, A., Liwicki, M., Fernandez, S., Bertolami, R., Bunke, H., & Schmidhuber, J. (2009). A novel connectionist system for unconstrained handwriting recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(5), 855-868.

[2] Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., & Wei, F. (2023). TrOCR: Transformer-based optical character recognition with pre-trained models. Proceedings of the AAAI Conference on Artificial Intelligence, 37(11), 13094-13102.

[3] Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., & Adam, H. (2019). Searching for MobileNetV3. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 1314-1324.

[4] Kumar, M., Sharma, R. K., & Sharma, V. (2020). A comprehensive review of OCR systems for Indian languages. Journal of King Saud University - Computer and Information Sciences, 32(4), 465-480.

[5] Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., & Zhou, M. (2020). LayoutLM: Pre-training of text and layout for document image understanding. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 1192-1200.

[6] Katti, A. R., Reisswig, C., Guder, C., Bruns, S., Gloger, J., & Wichmann, C. (2018). Chargrid: Towards understanding 2D documents. Proceedings of EMNLP 2018, 4459-4469.
[7] Smith, R. (2007). An overview of the Tesseract OCR engine. Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR), 629-633.
[8] Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., & Lee, H. (2021). What is wrong with scene text recognition model comparisons? Dataset and model analysis. IEEE/CVF ICCV, 4715-4723.

