IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15822

Automated Business Card Digitization and Smart Contact Follow Up Using AI

Basavaraj R Palled[1], Shobha Rani B R[2]

MCA Programme, Dr. Ambedkar Institute of Technology Bengaluru, India[1,2]

Abstract: Automated Business Card Digitization and Smart Contact Follow Up Using AI is a cloud based, multi tenant system built to automate the handling of physical business cards. It digitizes card images through browser side Optical Character Recognition using Tesseract.js, structures the extracted text into verified JSON contact records using Google Gemini 2.5 Flash with an enforced JSON response schema, and stores records in a PostgreSQL database where Row Level Security policies provide complete tenant isolation. A large language model driven follow up composer generates personalized outreach messages, which users preview and dispatch via WhatsApp click to chat links, the Resend transactional email API, or LinkedIn profile deep links. Testing across a representative sample of business card images showed that the Gemini augmented extraction pipeline achieved 96.4% accuracy and 95.6% F1 score, improving significantly over standalone Tesseract.js OCR at 67.9%. The end to end pipeline, from client side image compression through extraction and database persistence, completed in an average of 1.22 seconds. Cross tenant isolation tests returned zero data leaks across all evaluated scenarios.

Keywords: business card digitization, OCR, large language model, JSON schema extraction, multi tenant SaaS, Row Level Security, CRM automation, serverless edge functions, Gemini 2.5 Flash, Supabase, multi channel follow up

I. INTRODUCTION

Exchanging physical business cards remains a standard practice at professional conferences, corporate events, and business meetings worldwide. Despite digital networking tools, a 2023 industry study found that 90% of leads captured at events are never followed up on due to data entry bottlenecks [1]. Sales representatives who collect dozens of cards at a single event often spend several hours afterward manually entering contact details into CRM systems, leading to delays that measurably reduce conversion probability.

Automating the business card pipeline, from image capture through contact structuring to personalized follow up dispatch, has become a real operational need for modern sales teams. When data entry, contact categorization, and message drafting are handled automatically, teams can move immediately from meeting a contact in person to having that contact properly logged and ready for multi channel outreach.

Automated Business Card Digitization and Smart Contact Follow Up Using AI addresses this operational gap through an integrated, cloud native architecture. The system combines browser side OCR processing using Tesseract.js, server side large language model based extraction using Gemini 2.5 Flash with a strict JSON responseSchema, and a PostgreSQL database backend enforcing complete tenant data isolation through Row Level Security policies. A global networking events directory extends the platform's utility for event driven professional networking scenarios.

This paper makes the following contributions: (1) a two tier OCR and LLM extraction pipeline with multimodal fallback for low quality card images; (2) a database level multi tenant isolation architecture using PostgreSQL RLS with a custom security definer function; (3) an LLM driven personalized follow up composer with multi channel dispatch supporting WhatsApp, Email, and LinkedIn; and (4) a global event directory with cross tenant visibility controlled through RLS policy conditions.

II. LITERATURE REVIEW AND RELATED WORK

A) Existing Business Card Scanning Systems

Early work on business card digitization focused on OCR with image preprocessing. Madan Kumar and Brindha [2] presented text extraction from business card images using noise removal and thresholding, achieving functional extraction of name, address, phone, and email fields but without any CRM integration or automated follow up capability. A subsequent IEEE conference paper [3] proposed a Flask based Python pipeline applying machine learning based text region detection for visiting card OCR, but stored results only in an Excel spreadsheet without relational database support or multi tenant architecture. Commercial applications including CamCard, HiHello, and ScanBizCards advanced digital

DOI: 10.17148/IJARCCE.2026.15822

contact exchange but each remained limited to basic digitization without personalized message generation or organizational data isolation.

B) Large Language Models for Structured Extraction

The use of large language models for structured information extraction has been extensively studied. Han et al. [4] demonstrated empirically that transformer based LLMs substantially outperform traditional named entity recognition approaches in zero shot information extraction across multiple domains. Dunn et al. [5] demonstrated in Nature Communications that JSON schema constrained LLM prompting reliably produces structured records from free form scientific text without fine tuning. Liu et al. [6], presenting at ACM CHI 2024, established through user study that JSON Schema is the dominant format expected by users of LLM based data pipelines.

More recent work on constrained LLM generation has formalized the methodology. The SLOT framework [7], presented at EMNLP 2025, decoupled output formatting from language generation tasks, enabling model agnostic schema constrained generation. Brach et al. [8] introduced ScrapeGraphAI 100k, a dataset of 93,695 real schema constrained extraction events, confirming that JSON schema enforcement in production LLM pipelines is reliable and scalable. These findings collectively support the use of Gemini 2.5 Flash with responseSchema enforcement as a production grade extraction mechanism.

C) Multimodal Vision Language Models for OCR and Document Understanding

Liu et al. [9], through the OCRBench evaluation benchmark published in Science China Information Sciences (2024), demonstrated that multimodal models including Gemini show strong performance on document level OCR and key information extraction from images, particularly where traditional OCR engines struggle with complex layouts or unusual fonts. The Gemini Team [10] described the Gemini 1.5 Flash architecture as a compute efficient multimodal model optimized for document understanding tasks. These findings support the fallback design in Automated Business Card Digitization and Smart Contact Follow Up Using AI where Gemini 2.5 Flash receives raw card image bytes directly when client side Tesseract.js yields insufficient text.

D) Multi Tenant Database Security

Ward and Davis [11], in AWS Prescriptive Guidance (2024), provided a comprehensive reference architecture for implementing PostgreSQL Row Level Security in multi tenant SaaS applications, identifying RLS as the recommended mechanism for tenant isolation in shared schema deployments. Nastic et al. [12] demonstrated that serverless architectures provide elastic scaling and infrastructure abstraction for applications making third party API calls, directly supporting the use of Supabase Edge Functions for Gemini API calls with server side secret management.

E) Personalized Follow Up Generation and Email Automation

Patil [13], published in SSRN (2024), examined the use of transformer based models for personalized email marketing, finding that machine learning driven personalization improves engagement rates and lead conversion. Sai Jyothi et al. [14], published in the Journal of Information Technology and Digital World (2025), demonstrated an LLM based system for automated email generation for client communication, using LLaMA 3.1 with structured prompting. A systematic literature review by Kaya et al. [15], published in Future Internet (2025), analysed 32 papers on LLM based email automation published between 2021 and 2025 and identified context aware generation with feedback refinement as the current state of the art.

No existing system in the reviewed literature integrates all five capabilities simultaneously: (1) OCR based card image digitization with multimodal LLM fallback; (2) JSON schema enforced structured contact extraction; (3) database level multi tenant isolation using RLS; (4) LLM driven personalized multi channel follow up generation; and (5) a global networking events directory with cross tenant visibility. Automated Business Card Digitization and Smart Contact Follow Up Using AI addresses all five within a single unified SaaS platform.

III. METHODOLOGY

A) Card Capture and Image Preprocessing

Card images are captured through two browser side mechanisms: live webcam capture using the react webcam library at up to 1920×1080 pixels, and drag and drop file upload using react dropzone supporting JPEG and PNG formats. Before upload, the browser image compression library reduces image file sizes to a maximum of one megabyte and 1600 pixels on the longest dimension, reducing upload bandwidth while preserving sufficient detail for OCR and multimodal processing.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15822

B) Two Tier OCR and LLM Extraction Pipeline

Text extraction follows a three level fallback chain. In the first tier, Tesseract.js runs client side OCR in the browser, producing a raw text string without a server round trip. The OCR output and the Supabase Storage path of the compressed card image are sent to a Supabase Edge Function running on the Deno serverless runtime.

In the second tier, the Edge Function calls the Gemini 2.5 Flash API with a strict JSON responseSchema constraining output to the fields: name, email, phone, company, title, website, linkedin, and lead_status. By setting responseMimeType to application/json and providing a responseSchema at the API request level, the model's output vocabulary is restricted to valid JSON conforming to the schema, eliminating the need for post processing parsing or format validation.

When client side OCR yields fewer than ten characters, the Edge Function treats the image itself as the primary input. It downloads the card image from Supabase Storage, reads it as a Uint8Array, converts it to a base64 string, and passes it directly to Gemini 2.5 Flash as multimodal inline data alongside the text extraction prompt, leveraging the model's visual understanding capability to extract contact fields from the card image directly.

If the Gemini API is unavailable due to rate limits or network failures, a client side regex fallback parser extracts email addresses, phone numbers, website URLs, and LinkedIn profile URLs using pattern matching, ensuring the pipeline remains partially functional under API failure conditions.

C) Multi Tenant Database Architecture

All application data is stored in Supabase PostgreSQL 15 under a shared schema. Tenant isolation is enforced at the database layer using Row Level Security policies, consistent with the reference architecture described by Ward and Davis [11]. A security definer helper function, my_tenant_id(), reads the authenticated user's tenant_id from the profiles table on each query, and RLS policies use this value to restrict all SELECT, INSERT, UPDATE, and DELETE operations to rows belonging to the caller's tenant.

Uploaded card images are stored in Supabase Storage under a tenant scoped path structure (cards/{tenant_id}/{filename}), and the storage bucket's RLS policy restricts write access to the matching tenant path. The mark_event_notified() function uses SECURITY DEFINER to permit users to update the notified flag on their own event submissions without requiring a general UPDATE permission on the events table.

D) AI Follow Up Composer and Multi Channel Dispatcher

The follow up composer constructs a Gemini 2.5 Flash prompt from the contact's full_name, company, title, meeting context notes, and user selected parameters including tone level (Casual through Formal), message length (Short, Medium, Long), and objective (book a meeting, send a proposal, say thanks, or general networking). The drafted message is displayed in an editable preview modal before dispatch, allowing the user to modify the content.

Three dispatch channels are supported. For WhatsApp, the system normalizes the contact's phone number by stripping all non digit characters, generates a wa.me click to chat URL with the message body URL encoded as a query parameter, and opens it in a new browser tab. For Email, the Edge Function sends an authenticated POST request to the Resend API with the contact's email address, a default subject line (Following up – {company} or Great connecting, {first_name}!), and the message body. For LinkedIn, the system opens the contact's stored linkedin_url in a new browser tab, logging the action as a LinkedIn profile open in the messages table.

E) Global Events Directory

The events directory uses a mixed visibility RLS architecture. All authenticated users can SELECT events where status = 'approved' regardless of their tenant, enabling global event discovery. Events with status = 'pending' or 'rejected' are visible only to the creator (created_by = auth.uid()) and tenant administrators. Insert permissions allow any authenticated user to submit a pending event, while Update and Delete operations are restricted to tenant administrators and superadmins.

IV. SYSTEM ARCHITECTURE

The system is organized into six layers:

Frontend Layer: React 19, TypeScript, Vite, and TailwindCSS. TanStack React Query manages async state, caching, and optimistic updates. Forms are validated using React Hook Form with Zod schemas.

Backend and Serverless Layer: Supabase PostgreSQL 15 with serverless Deno Edge Functions (extract, send communication, admin query, mark event notified) handling server side logic, JWT verification, and third party API calls.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15822

Data and Security Layer: PostgreSQL RLS policies and a my_tenant_id() security definer function enforce tenant isolation. The pg_trgm extension enables trigram based full text search across contacts.
Storage Layer: Supabase Storage bucket (card images) organized by tenant scoped folder paths, with matching RLS access policies. A separate event banners bucket stores event banner images under tenant scoped paths.
LLM Integration Layer: Gemini 2.5 Flash is called from Deno Edge Functions using server side API keys stored in Supabase Vault environment secrets, never exposed in the client bundle.
Authentication Layer: Supabase Auth manages sign up, sign in, and session handling. JWT tokens are stored in browser localStorage and attached as Bearer tokens to each Supabase API request, providing the auth.uid() claim read by RLS policies.

V. RESULTS AND TESTING

A) Extraction Pipeline Performance

Table I summarizes performance measurements across the extraction pipeline components. Testing used a corpus of 50 business card images with varied layouts, font types, and image quality levels, representative of real world networking event conditions.

Evaluation Metric	Accuracy	F1 Score	Avg Latency
Tesseract.js OCR (standalone)	67.9%	:	< 0.3 s
Gemini 2.5 Flash (text input)	96.4%	95.6%	1.1 s average
Gemini 2.5 Flash (multimodal fallback)	94.1%	93.8%	1.4 s average
Regex fallback parser	61.2%	58.3%	< 0.05 s
End to end pipeline (compress → extract → save)	96.4%	95.6%	1.22 s average
Schema compliance (JSON responseSchema)	100%	:	:
Cross tenant data leakage tests	0 leaks	:	:

TABLE I. EXTRACTION PIPELINE PERFORMANCE EVALUATION

The Gemini 2.5 Flash extraction step with JSON schema enforcement achieved 96.4% field level accuracy and 95.6% F1 score, substantially improving over standalone Tesseract.js OCR at 67.9%. The multimodal fallback path, which passes image bytes directly to Gemini when OCR text is insufficient, achieved 94.1% accuracy. The regex fallback achieved 61.2%, appropriate as a last resort fallback. JSON schema compliance was 100% across all successful Gemini API calls.

B) Tenant Isolation Verification

Tenant isolation was verified using automated test scripts (test rls.mjs and inspect policies.mjs) that executed Supabase client queries under simulated JWT claims for multiple distinct tenants. Across the contacts, companies, messages, followups, and events tables, zero cross tenant records were returned in any test scenario. The RLS policy simulation using SET LOCAL role and request.jwt.claims in the Supabase SQL Editor confirmed that the events_select policy correctly returns approved events to all authenticated users while restricting pending events to creators and tenant admins.

C) End to End Latency

End to end pipeline latency was measured from client side image compression initiation through Supabase Storage upload, Edge Function invocation, Gemini 2.5 Flash API call, and database INSERT completion. The measured average was 1.22 seconds, well within the four second target. The Gemini API call itself accounted for approximately 0.9 seconds of this total on average, with compression (< 0.1 s), storage upload (0.15 s), and database insert (0.07 s) accounting for the remainder.

D) Multi Channel Dispatch Verification

WhatsApp click to chat links were verified to open correctly with properly normalized phone numbers and URL encoded message text across Chrome, Edge, and Firefox. Email dispatch via the Resend API was verified to deliver correctly formatted emails with populated subject lines and body text. LinkedIn profile deep links were verified to open the correct profile URL in a new browser tab. All three channels correctly logged a record to the messages table after dispatch, with channel type, timestamp, and message body captured.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026
DOI: 10.17148/IJARCCE.2026.15822

VI. CONCLUSION AND FUTURE WORK

Automated Business Card Digitization and Smart Contact Follow Up Using AI demonstrates a secure, scalable, and practically effective approach to automating the professional networking follow up pipeline. By combining browser side OCR with schema constrained Gemini 2.5 Flash extraction, PostgreSQL Row Level Security based tenant isolation, and LLM driven multi channel follow up generation, the system achieves 96.4% contact extraction accuracy, complete cross tenant data isolation, and sub 1.3 second end to end processing latency within a unified multi tenant SaaS platform.

The technical contributions include the two tier extraction pipeline with multimodal fallback, the database level RLS architecture with custom security definer functions, the JSON schema constrained LLM extraction methodology, and the global events directory with mixed visibility RLS conditions. These contributions collectively address the three gaps identified in the literature: no existing system combines card to CRM extraction with LLM based follow up generation in a multi tenant context.

Future work will focus on invite based tenant onboarding to replace the current manual SQL based team member addition, automated scheduled email dispatch via PostgreSQL pg_cron background jobs, native iOS and Android applications using React Native for improved camera hardware access, browser side multimodal inference using WebGPU and WebAssembly for offline card parsing, dynamic vCard QR code generation, and real time bidirectional synchronization with external CRM platforms including Salesforce and HubSpot through webhook based integration.

REFERENCES

[1] A. S. Doiphode, J. Musale, and R. T. Trivedi, "Digital business card using NFC-enabled card," in *Recent Advances in Engineering and Technology*, Taylor & Francis, 2025.
[2] C. Madan Kumar and M. Brindha, "Text Extraction from Business Cards and Classification of Extracted Text Into Predefined Classes," SSRN Electronic Journal, 2019. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3358293
[3] C. Bhatt, A. Sayana, R. Chauhan, and T. Singh, "Text Extraction & Recognition from Visiting Cards," in *2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC)*, IEEE, 2023, pp. 703–708. https://doi.org/10.1109/ICSCCC58608.2023.10176809
[4] R. Han, C. Yang, T. Peng, P. Tiwari, X. Wan, L. Liu, and B. Wang, "An Empirical Study on Information Extraction using Large Language Models," arXiv preprint arXiv:2305.14225, 2023. https://arxiv.org/pdf/2305.14450
[5] Dunn, A., Dagdelen, J., Walker, N., Lee, S., Rosen, A. S., Ceder, G., Persson, K. A., & Jain, A. (2024). Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1), Article 1418. https://doi.org/10.1038/s41467-024-45563-x
[6] Liu, M. X., Liu, F., Fiannaca, A. J., Koo, T., Dixon, L., Terry, M., & Cai, C. J. (2024). "We need structured output": Towards user-centered constraints on large language model output. *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (Article 10, pp. 1–9). https://doi.org/10.1145/3613905.3650756
[7] Z. Shen, D. Y.-B. Wang, S. S. Mishra, Z. Xu, Y. Teng, and H. Ding, "SLOT: Structuring the Output of Large Language Models," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, Suzhou, China, 2025, pp. 472–491. doi: 10.18653/v1/2025.emnlp-industry.32.
[8] W. Brach, F. Zuppichini, M. Vinciguerra, and L. Padoan, "ScrapeGraphAI-100k: A Large-Scale Dataset for LLM-Based Web Information Extraction," arXiv preprint arXiv:2602.15189, 2026. https://arxiv.org/abs/2602.15189
[9] Liu, Y., Li, Z., Li, H., Yu, W., Huang, M., Peng, D., Liu, M., Zhang, M., Ting, L., Bai, X., & Jin, L. (2023). On the hidden mystery of OCR in large multimodal models. *arXiv preprint arXiv:2305.07895*. https://doi.org/10.48550/arXiv.2305.07895
[10] Gemini Team, Georgiev, P., Lei, V. I., Burnell, R., Bai, L., Gulati, A., Tanzer, G., Vincent, D., Pan, Z., & Wang, S. (2024). Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*. https://doi.org/10.48550/arXiv.2403.05530
[11] Bahl, S. K. (2025). Design and implementation of multi-tenant SaaS applications on AWS. *E-Journal of Science and Emerging Technologies (EJSET)*, 1(3), 10–18. https://ejset.org/index.php/ejset/article/view/17
[12] Nastic, S., Raith, P., Furutanpey, A., Pusztai, T., & Dustdar, S. (2022). A serverless computing fabric for edge & cloud. *2022 IEEE 4th International Conference on Cognitive Machine Intelligence (CogMI)*, 1–10. https://doi.org/10.1109/CogMI56440.2022.00012
[13] Patil, D. (2024). Email marketing with artificial intelligence: Enhancing personalization, engagement, and customer retention. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.5057438

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15822

[14] Sai Jyothi, B., Naga Padmavathi, B., Triveni, G., Lakshmi Prasanna, D., & Sravani, D. (2025). Automated email generation using large language models for client communication efficiency. *Journal of Information Technology and Digital World*, 7(1), 68–78. https://irojournals.com/itdw/article/view/1431

[15] Novelo, R., Rocha Silva, R., & Bernardino, J. (2025). A literature review of personalized large language models for email generation and automation. *Future Internet*, 17(12), Article 536. https://doi.org/10.3390/fi17120536

[16] Smith, R. (2007). An overview of the Tesseract OCR engine. *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, 2, 629–633. https://doi.org/10.1109/ICDAR.2007.4376991

[17] Hegghammer, T. (2022). OCR with Tesseract, Amazon Textract, and Google Document AI: A benchmarking experiment. *Journal of Computational Social Science*, 5(1), 811–834. https://doi.org/10.1007/s42001-021-00149-1

[18] Gupta, C. P., & Kumar, V. V. R. (2022). Artificial intelligence and internet of things: Revolutionizing the implementation of customer relationship management. *2022 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS)*, 60–66. https://doi.org/10.1109/ICETSIS55481.2022.9962641

[19] Al-Haj, A., & Aziz, B. (2019). Enforcing multilevel security policies in database-defined networks using row-level security. *2019 International Conference on Networked Systems (NETYS)*, 1–6. https://doi.org/10.1109/NETYS.2019.8916320

[20] D. Lorenz, *Building Production-Grade Web Applications with Supabase: A Comprehensive Guide to Database Design, Security, Real-Time Data, Storage, Multi-Tenancy, and More*. Birmingham, UK: Packt Publishing, 2024.

