

RAMAN THAKUR

Senior AI Engineer | Full-Stack Developer | Systems Architect

raman1801thakur@gmail.com | +917880211396 | Delhi, India
[GitHub](#) | [LinkedIn](#)

PROFESSIONAL SUMMARY

Senior AI Engineer with 2+ years of experience building production-ready AI systems, enterprise applications, and distributed platforms. Specialized in developing end-to-end solutions spanning computer vision, natural language processing, cybersecurity, financial systems, and multi-agent architectures. Proven track record at Labellerr building SDKs, RAG systems, and documentation infrastructure. Expert in deploying scalable AI solutions using cutting-edge frameworks including LangGraph, CrewAI, and Model Context Protocol.

TECHNICAL SKILLS

AI & Machine Learning

- **LLMs & Frameworks:** GPT, Claude, Gemini, Qwen, Llama, LangChain, LlamaIndex, CrewAI, AutoGen
- **Computer Vision:** YOLOv8, SAM, SAM-2, OWL-ViT, DETR, CVAT, Segment Anything
- **ML Frameworks:** PyTorch, TensorFlow, Scikit-Learn, Transformers, Sentence Transformers

Programming & Development

- **Languages:** Python (Advanced), TypeScript, Rust, SQL, JavaScript, Dart
- **Frameworks:** FastAPI, React, Flutter, Next.js, Actix-web
- **Databases:** PostgreSQL, MongoDB, SQLite, Vector DBs (ChromaDB, Pinecone, FAISS, Qdrant)

Infrastructure & DevOps

- **Cloud & Deployment:** Docker, Kubernetes, GitHub Actions, AWS, GCP
- **Tools:** Git, Linux, Playwright, Redis, Nginx

Specialized Domains

- Cybersecurity, Blockchain, Federated Learning, Multi-Agent Systems, RAG Pipelines, Web Scraping

WORK EXPERIENCE

AI Growth Engineer | Labellerr

May 2024 – October 2025

SDK Development & Integration

- Enhanced Python SDK with project creation methods and developer-friendly abstractions for data annotation workflows
- Improved API integration serving 500+ developers
- Contributed to core library improvements and documentation infrastructure

RAG System Architecture

- Designed and implemented RAG chatbot for Q&A over Labellerr website, documentation, and SDK
- Enabled real-time knowledge retrieval with LLMs for intelligent support automation
- Achieved 92% retrieval accuracy on technical queries using Qwen embeddings and LangChain

Technical Documentation

- Migrated comprehensive documentation from Notion to Theneo (MDX-based)
- Reduced support queries by ~35% through improved developer onboarding
- Established documentation standards for technical accuracy and accessibility

Model Evaluation & Testing

- Conducted comprehensive fine-tuning, testing, and evaluation of 200+ LLMs, VLMs, and CV models
- Built evaluation frameworks and benchmarking pipelines for model selection
- Assessed performance metrics, inference capabilities, and deployment feasibility

Technical Research

- Authored 150+ technical articles covering LLM architectures, VLM applications, and AI system design
- Focused on practical implementation, performance optimization, and deployment strategies

FEATURED PROJECTS

Enterprise AI & Security Systems

NEXUS GUARD - Enterprise AI Cybersecurity Platform *Python, FastAPI, PyTorch, Blockchain, Federated Learning*

- Built comprehensive AI-powered cybersecurity platform with multi-modal threat detection
- Implemented federated learning network for privacy-preserving threat intelligence sharing
- Developed blockchain-based immutable audit system with smart contract automation
- Integrated LSTM behavioral analysis, CNN signature detection, and graph neural networks
- Achieved sub-second threat detection with automated response orchestration

Universal Web Scraper *Python, Playwright, FastAPI, Transformers, FAISS*

- Developed production-ready web scraping framework as open-source alternative to Firecrawl
- Implemented JavaScript rendering, OCR (Tesseract), and audio transcription (Whisper)
- Built RAG-ready vector storage with ChromaDB, FAISS, and Qdrant support
- Created intelligent rate limiting and anti-detection mechanisms
- Supports 10,000+ requests/second with concurrent processing

Pentesting AI Agent *Python, Aircrack-ng, Hashcat, Natural Language Processing*

- Created AI-powered penetration testing automation tool with natural language interface
- Automated wireless network analysis, password cracking, and security assessments
- Built multi-tool integration (nmap, hydra, sqlmap) with intelligent workflow orchestration
- Developed web interface and REST API for programmatic access

Computer Vision & Automation

Viseon - Computer Vision Platform *Python, FastAPI, YOLO, CVAT, MLflow, Docker*

- Developed comprehensive end-to-end computer vision platform for data management, model training, and inference
- Built CVAT integration for annotation workflows with project-based dataset organization
- Implemented YOLO-based training pipeline with MLflow experiment tracking
- Created high-performance inference server with FastAPI and object tracking system (ByteTrack, DeepSORT, BoT-SORT)
- Designed REST API for external integrations with real-time monitoring capabilities

Agentic Vision - Interactive Multi-Model CV System *Python, FastAPI, OWL-ViT, SAM, PyTorch, Gradio*

- Engineered intelligent vision system combining OWL-ViT (open-vocabulary detection) with SAM (segmentation)
- Built FastAPI backend with Gradio web interface for real-time vision tasks
- Enabled complex multi-step instructions: detect objects → segment boundaries → apply transformations
- Impact: 85% reduction in processing time with production-ready deployment

Development Tools & Automation

Advanced Coding and Testing Agent *Python, AI Agent Orchestration, Multi-Framework Support*

- Built comprehensive AI agent handling complete software development lifecycle from natural language queries
- Implemented automated planning, development, testing, deployment, and maintenance workflows
- Supported 8 project types: Web apps (React + FastAPI), ML pipelines, API services, data analysis, desktop/mobile apps, automation, CLI tools
- Integrated feasibility assessment, library discovery, and comprehensive testing across multiple phases
- Generated production-ready code with best practices, complete documentation, and automated testing

Intelligent Agents & Assistants

S.I.Y.A Enhanced - AI Assistant *Python, Qwen, Transformers, SQLite, FastAPI*

- Built Jarvis-style AI assistant with sub-100ms response times
- Implemented intelligent time-based greetings, real-time data fetching, and general web search
- Developed activity monitoring system with smart work suggestions
- Created persistent notes & reminders system with SQLite backend
- LoRA fine-tuning pipeline for response customization

Phone Assistant Agent *Python, FastAPI, Twilio, Transformers, Whisper*

- Developed intelligent phone assistant for automated call handling and document processing
- Integrated Google Calendar/Outlook for appointment scheduling
- Implemented voice transcription using Whisper and smart reply generation
- Built support for multiple AI models (OpenAI, Gemini, Claude, local models)

Research Paper Explainer *Python, FastAPI, React, LangGraph, FAISS*

- Created multi-agent AI system for comprehensive research paper analysis
- Implemented dual explanation levels (simple and technical) with diagram analysis
- Built paper comparison engine with semantic search capabilities
- Developed vector-based similarity search across research papers
- Real-time processing with WebSocket-based progress updates

Trading & Financial Systems

Live Cryptocurrency Trading System *Python, Binance API, SQLite, Multi-Strategy Portfolio*

- Developed production-ready automated Bitcoin trading system with real-time monitoring
- Implemented four proven strategies: Grid Trading (40%), Momentum (25%), Funding Arbitrage (25%), DCA (10%)
- Built dynamic strategy selection system optimizing allocations based on performance
- Created comprehensive dashboard for live profit tracking and trade analytics

- Achieved positive returns across 90-day backtest in bearish market (-18.62%)

Messaging & Communication

Secure Messaging App *Rust, Flutter, libp2p, Signal Protocol*

- Developed P2P encrypted messaging application with mesh networking capabilities
- Implemented end-to-end encryption using AES-256-GCM and Signal Protocol
- Built Bluetooth mesh networking for offline communication
- Created cross-platform mobile app with Flutter (iOS, Android, Web)
- Integrated AI-powered smart replies using local LLMs

CORE COMPETENCIES

AI Model Development

- Model Context Protocol (MCP): File systems, API integrations, custom protocols
- RAG Systems: LangChain, LlamaIndex, vector databases
- Fine-tuning: LoRA, QLoRA, model adaptation
- Evaluation: LLM-as-Judge systems, benchmarking pipelines

System Architecture

- Multi-agent orchestration with CrewAI and LangGraph
- Microservices architecture with FastAPI
- Real-time systems with WebSocket and async processing
- Distributed systems and federated learning

Development Practices

- Test-driven development with pytest
- CI/CD with GitHub Actions
- Docker containerization and Kubernetes orchestration
- API design and documentation (OpenAPI/Swagger)

RESEARCH CONTRIBUTIONS

Wheat Leaf Disease Detection Using YOLOv8 and Conditional GAN

- Generated synthetic diseased wheat leaf images using Conditional GAN
- Implemented YOLOv8 for state-of-the-art disease classification
- Achieved 92% detection accuracy with minimal training data

Optimized Mask RCNN for Liver Tumor Detection

- Applied Mask R-CNN for precise tumor boundary delineation
- Implemented custom loss functions for medical imaging
- Achieved 88% IoU score on clinical datasets

FRF-BiLSTM: DDoS Attack Detection

- Designed federated learning framework for distributed attack detection
- Implemented BiLSTM with feature optimization
- Demonstrated privacy-preserving ML in security applications

EDUCATION

Bachelor of Technology - Computer Science and Engineering

Kalinga Institute of Industrial Technology (KIIT)

CGPA: 7.52 | Graduated: 2024

AVAILABILITY

Open to:

- Full-time AI Engineer / Senior Developer roles
- Contract AI engineering work (hourly or project-based)
- Freelance consulting for AI systems and architecture
- Open-source collaboration on AI/ML projects

Timezone: IST (UTC+5:30) | Available for global remote teams

Work Mode: Flexible (remote-first, can travel for important projects)

