

JITEN RAI

+977 9860270335 | info@jitenrai.com.np | [LinkedIn](#) | [Portfolio](#) | [Github](#)

PROFILE SUMMARY

Aspiring AI/ML engineer with hands-on experience across computer vision, deep learning, and applied AI, gained through two internships and multiple end-to-end projects. Skilled in building and deploying full-stack ML systems, from CNN/YOLO-based real-time detection pipelines to RAG-powered developer tools - using Python, TensorFlow, FastAPI, and Docker. Experienced in the complete ML lifecycle: data preprocessing, model training and calibration, API integration, and production deployment, with a strong focus on reproducibility, fairness-aware evaluation, and system architecture.

EDUCATION

Texas College of Management and IT

Bachelor of Information Technology (Hons)

Mitrapark, Kathmandu

2022 – 2026

WORK EXPERIENCE

AI Intern

Nov 2025 - Jan 2026

Frost Digital Ventures

Lampati, Kalanki

- Completed multiple end-to-end weekly projects spanning EDA, classical ML, CNNs, LLMs, and RAG - including a Telco churn pipeline (XGBoost, SMOTE, nested CV) achieving high recall with cost reduction via segment-specific fairness thresholds, and a CIFAR-10 vision pipeline (ResNet18, MobileNetV2, GradCAM, FGSM robustness testing) reaching 90%+ accuracy.
- Prepared comprehensive task submissions including calibration, ablation studies (temperature scaling, RandAugment vs. Mixup/CutMix), LoRA-PEFT fine-tuning (FLAN-T5, RoBERTa, DistilBERT achieving 90% accuracy with 0.70% trainable parameters), evaluation using MRR and nDCG metrics, and knowledge-graph construction with NetworkX and Neo4j (centrality, PageRank, node2vec-based classification).
- Built a BM25 + FLAN-T5-XL RAG pipeline benchmarked against prompt-engineered and fine-tuned baselines, deployed multiple production-style applications using FastAPI, Streamlit, and Docker, and implemented experiment tracking with MLflow and Weights Biases across all projects.
- Developed a complete AI-powered developer Copilot application, combining semantic code search (HyDE/RAG), PR summarization, and local embedding + external LLM architecture and delivered it as a production-ready VS Code extension.

Computer Vision Intern

May 2025 - Aug 2025

Danson Solutions

Gairidhara, Kathmandu

- Led a team in building an interactive, real-time computer vision game that combined camera-based detection, multi-threaded architecture, and game rendering, coordinating development across model integration, calibration, and system synchronization to deliver a functioning end-to-end demo.
- Developed multiple computer vision and deep learning mini-projects, including a CNN-based sketch recognition system (QuickDraw) achieving 86.5 validation accuracy and a real-time YOLO11-based object detection game, applying skills across model training, image preprocessing, and inference optimization.
- Applied Generative AI concepts and built FastAPI-based backend endpoints to serve trained models for real-time inference, handling request processing, model loading, and structured API responses to support interactive frontend applications.

PROJECTS

Projector Camera Interactive Balloon Pop Game | [LINK](#) | Python, Pygame, OpenCV, Ultralytics YOLO, ONNX

- Built a real-time interactive game where a camera detects physical balloons projected on a wall and auto-pops them via synthetic mouse clicks, combining computer vision, real-time inference, and game rendering into a single coordinated pipeline.
- Trained a custom YOLO11-nano model for single-class balloon detection on 2000+ frames extracted from recorded gameplay footage, exporting to ONNX (9.46M params) for CPU inference; chose YOLO11-nano for its C3k2 blocks and self-attention backbone, which gave better feature extraction on a small dataset than a pure CNN nano architecture while staying within CPU budget.
- Implemented interactive 4-point camera-to-screen calibration using homography transforms, with validation checks (aspect ratio, minimum point distance, quadrilateral ordering) and persisted calibration data that auto-invalidates on screen resolution change.
- Refactored inline game logic into dedicated modules (balloon physics, visual effects, centralized config) and removed import-time side effects, improving testability and decoupling screen-dimension logic from module imports.
- Architected a four-thread system (camera capture, YOLO inference, audio, and main game loop) synchronized via RLock for shared state, Event-based frame-ready signaling, and a bounded queue with latest-value semantics for detection buffering, keeping camera capture at 30 FPS without blocking rendering or audio.

Developer Copilot: AI-Powered RAG and PR Summarization | [LINK](#) | Python, ChromaDB, Groq, RAG, VS Code Extension

- Built DevCopilot v2, an AI developer tool combining semantic code search (HyDE/RAG) and PR summarization, using a hybrid architecture with local embeddings for privacy and external LLM APIs (Groq llama-3.3-70b) for generation.

- Implemented dual search pipelines - HyDE (LLM generates hypothetical code, embedded and matched via vector search) and RAG (retrieves relevant chunks, then generates contextual explanations) - alongside a repository indexer using tree-sitter for AST-based function/class extraction, with embeddings via SentenceTransformers (Jina v2 Code, 768-dim) stored in a per-repository ChromaDB (HNSW cosine index).
- Developed an automated PR summarization pipeline that extracts git diffs and commit history, classifies atomic code changes per file, and generates structured summaries via LLM calls with retry logic (exponential backoff, rate-limit handling).
- Designed a VS Code extension (TypeScript) that bridges to a Python CLI (Typer + Rich) via child-process spawning, parsing Rich-formatted terminal output into structured UI data, with a 3-tier API key resolution flow (OS keychain, environment variable, settings) for secure credential handling.
- Iterated the architecture from Jupyter notebook prototyping to functional scripts to a class-based CLI and extension, reflecting a structured, staged development approach.

QuickDraw Game - AI-Powered Drawing Recognition | LINK | *Python, FastAPI, TensorFlow, Vanilla JS, Keras, Render*

- Preprocessed raw QuickDraw .npy data by reshaping 784-length vectors into 28x28 images, normalizing pixel values to 0-1, capping samples at 10k per class across 32 classes, and applying a 90/10 stratified split with a fixed seed for reproducibility.
- Designed a compact 2-layer CNN (32 and 64 filters) suited to the low visual complexity of sketch data, using max-pooling and progressively narrowing dense layers (512 to 128 to 32) with dropout to prevent overfitting.
- Trained the model using the Adam optimizer and categorical crossentropy loss, achieving 86.5 validation accuracy within 3 epochs.
- Integrated the trained .keras model into a FastAPI backend with async inference and endpoints for drawing recognition, random object selection, and model metadata; connected it to a Vanilla JS canvas frontend for real-time interaction.
- Deployed the full-stack application on Render **APPLICATION LINK**, configuring backend and frontend for production hosting and public accessibility.

CERTIFICATIONS

- AI Internship Experience Letter | LINK | Frost Digital Ventures — Jan 2026**
- Computer Vision Internship Letter | LINK | Danson Solutions — Aug 2025**
- Deep Learning Specialization | LINK | Coursera — Feb 2025**
- Sequence Models | LINK | DeepLearning.AI — March 19, 2025**
- Convolutional Neural Network | LINK | DeepLearning.AI — March 10, 2025**
- Structuring Machine Learning Projects | LINK | DeepLearning.AI — March 4, 2025**
- Improving Deep Neural Networks: Hyperparameter Tuning, Regularization | LINK | DeepLearning.AI — February 27, 2025**
- Neural Networks and Deep Learning | LINK | DeepLearning.AI — February 13, 2025**

SKILLS SUMMARY

- Programming:** Python, JavaScript, Data Structures & Algorithms
- ML/DL Libraries:** TensorFlow, Keras, PyTorch (basic), Scikit-learn, XGBoost, Hugging Face Transformers
- Domains:** Computer Vision, Deep Learning, NLP, RAG Systems, LLM Fine-Tuning (LoRA/PEFT), Machine Learning
- Computer Vision:** OpenCV, YOLO (Ultralytics), ONNX Runtime, Image Preprocessing & Augmentation
- NLP & LLMs:** Prompt Engineering, RAG (HyDE/BM25), SentenceTransformers, ChromaDB, Groq API
- Backend & Frameworks:** Django, Django REST Framework, FastAPI, HTMX
- Databases:** PostgreSQL, SQLite, ChromaDB (vector search)
- Deployment & Tools:** Docker, Render, MLflow, Weights & Biases, Git, GitHub Actions (CI)
- Soft Skills:** Problem-Solving, Data Analysis, Project Management, Team Leadership, Collaboration

ADDITIONAL INFORMATION

- Languages:** English (Fluent), Nepali (Native)
- Awards/Activities:**
 - Winner, Texas Dev Rumble 2025 - One Day Code Competition, Collaborated with a team developed an end-to-end web application on virtual study rooms with session based context level understanding chatbot feature
 - Winner, Texas Hackathon 2025 - Volume I, Collaborated with my team to develop a virtual assistant for visually impaired students
 - Robotics Expo - Sep 2025, Presented a group project on a Remote-Controlled IR Ray Sensor Light built using Arduino and IoT concepts

REFERENCES

References available upon request.