

# SAHIL KUMAR

+91 8168560627 | sahilkumar.dev

✉ sahil115jangid@gmail.com in linkedin.com/in/sahilkumar119 🌐 github.com/SahilJangra115

## Summary

---

AI/ML Engineer with hands-on experience building and deploying deep learning models for computer vision, NLP, and multimodal AI systems. Built end-to-end projects including an autonomous robot with concurrent real-time vision/voice/hardware loops, a TinyML watchdog on ESP32-S3, a from-scratch multimodal RAG system, and an AI misinformation detector. Proficient in PyTorch, CUDA, FastAPI, and embedded systems (FreeRTOS, micro-ROS). Strong builder who ships tested, production-hardened code.

## Technical Skills

---

**Programming Languages:** Python, Go, C/C++, CUDA, Java, JavaScript, TypeScript, SQL

**ML/AI Frameworks:** PyTorch, OpenCV, Hugging Face Transformers, LangChain, LangGraph, LlamaIndex, Ollama, ONNX Runtime

**Speech & NLP:** Whisper / faster-whisper, WhisperX forced alignment, pyannote diarization, sentence-transformers, LoRA/PEFT fine-tuning

**Vector Stores & Databases:** FAISS, Qdrant, Chroma, Pinecone, PostgreSQL, SQLite, Redis

**Backend & App Frameworks:** FastAPI, Flask-SocketIO, Flutter, Riverpod, Pydantic, Pytest

**Embedded & Robotics:** ESP-IDF, FreeRTOS, micro-ROS / ROS 2, Arduino, ESP32 / ESP32-S3

**Tools & Platforms:** Git, Docker, Kubernetes, GitLab CI, GitHub Actions, UV, AWS Lambda, Linux, VS Code, GNU Make

## Personal Projects

---

### StreamForge: Long-Form Cricket Audio to Structured, Searchable Content | *Go, Python, Qdrant* 2026

- Built the ingest and orchestration control plane in Go: ffmpeg HLS segmenting, a Redis-backed work queue with worker leases, idempotent re-runs and retry/backoff, dispatching to a pool of Python GPU workers so one hour of podcast audio finishes end to end in ~7 minutes on a single consumer GPU.
- Ran ASR on hour-long, multi-speaker Hinglish audio: faster-whisper large-v3 with VAD-packed chunks, WhisperX forced alignment for word-level timestamps, and pyannote diarization to attach speaker turns.
- Cut word error rate on player names and cricket terminology from 31% to 12% with a cricket lexicon bias pass plus fuzzy entity repair against a curated gazetteer.
- Post-processed transcripts with LLMs into typed material: disfluency cleanup, topic-boundary chaptering, host/guest role labels, and extraction of claims, predictions and debate takes as Pydantic-validated JSON with schema-failure retries.
- Built grounded retrieval over the transcripts: speaker-turn-aware chunking, BGE embeddings in Qdrant, hybrid BM25 + dense search with cross-encoder rerank; every answer cites a word-level timestamp and abstains when no chunk clears the evidence threshold.
- Packaged each stage as a FastAPI microservice with UV-managed environments, containerized and deployed to Kubernetes through GitLab CI, with a GPU/CPU split per stage and cost per audio-hour tracked per run.
- Measured quality like production: a hand-labeled golden set of 40 cricket clips (~6 hours) scoring WER, cricket-entity WER, diarization error rate, retrieval recall@5 and grounding faithfulness, with failures bucketed by cause and a CI gate that fails the build on a >3% regression.

### NOIR: Autonomous Multimodal Robot Assistant | *Python, PyTorch, VLM, Ollama, Flutter* 2025

- Built a mobile robot running three concurrent real-time loops (vision, voice, hardware control) in Python, with shared state behind a reentrant lock and deep-copied snapshots so no loop holds a mutable reference into another.
- Shipped a fully local voice pipeline (openWakeWord hotword, Faster-Whisper STT on CUDA, Gemma via Ollama, Piper TTS) that also accepts microphone audio streamed from the phone app.
- Stabilized YOLOv8n person tracking with an EMA filter so bounding-box jitter never reaches the motors, and trained a MobileNetV2 classifier for line following.
- Hardened every hardware action: Pydantic-validated LLM output, allow-listed motor commands with bounded durations, and ~2,000 lines of unit, integration, e2e, and resilience tests.
- Wrote the ESP32 firmware in C on FreeRTOS (ESP-IDF v6) driving an L298N H-bridge over TCP, plus a Flutter app for live telemetry and manual control over WebSocket.

### NyxCore: On-Device TinyML Watchdog for ROS 2 | *C, ESP32-S3, micro-ROS, PyTorch* 2026

- Built an independent watchdog on a separate ESP32-S3 that monitors a robot's DDS streams and keeps reporting even when the main compute node stalls or crashes.
- Ran anomaly inference in pure C with no ML runtime: a Mahalanobis distance scorer (explicit 4x4 matrix multiply) and a sub-400-byte autoencoder trained in PyTorch, weights baked into firmware as static arrays.
- Scored each topic (cmd\_vel, scan, odom) independently into a worst-case health level, with a 3-second silence watchdog catching dead publishers the statistical scores miss.
- Shipped it as a managed micro-ROS lifecycle node on FreeRTOS publishing diagnostic\_msgs at 1 Hz, validated by a 9-script Python fault-injection harness (rate drops, jitter, message loss, silent topics).

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| <b>image-N: From-Scratch Multimodal RAG System</b>   <i>Python, FAISS, CLIP, Ollama, FastAPI</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | <b>2026</b> |
| <ul style="list-style-type: none"> <li>Built a multimodal Retrieval-Augmented Generation system from scratch in pure Python with zero framework glue: FAISS vector search, CLIP image retrieval, local Ollama LLMs, and a FastAPI service.</li> <li>Implemented semantic search over text and images using CLIP embeddings and FAISS indexing for fast approximate nearest-neighbor retrieval.</li> <li>Designed a clean API layer with FastAPI serving retrieval and generation endpoints, supporting both text-to-text and image-to-text queries.</li> </ul>                                                                                                                                                                                                                                                                       |             |
| <b>Holmes: AI Content Provenance &amp; Misinformation Detector</b>   <i>PyTorch, DeBERTa-v3, ONNX</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | <b>2026</b> |
| <ul style="list-style-type: none"> <li>Fused four independent detection signals (DeBERTa-v3 text classifier, C2PA/EXIF metadata, SynthID watermark detection, image forensics) into a calibrated confidence score with uncertainty estimation.</li> <li>Optimized the model pipeline with ONNX Runtime for low-latency inference and deployed the service on AWS Lambda for serverless scalability.</li> <li>Built the backend API with FastAPI, handling concurrent classification requests with structured input validation via Pydantic.</li> </ul>                                                                                                                                                                                                                                                                               |             |
| <b>HealthByte: Personalized Health &amp; Nutrition Engine</b>   <i>PyTorch, EfficientNet-B0, SQLite, Google Fit</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | <b>2026</b> |
| <ul style="list-style-type: none"> <li>Trained an EfficientNet-B0 food classification model with uncertainty-banded calorie estimation, providing statistically honest confidence intervals on predictions.</li> <li>Built a biometric time-series pipeline using SQLite for persistent storage and Google Fit OAuth integration for real-time health data ingestion.</li> <li>Implemented statistically rigorous correlation analysis between nutrition intake and health metrics.</li> </ul>                                                                                                                                                                                                                                                                                                                                       |             |
| <b>Path Tracer: From-Scratch C++17 and CUDA Renderer</b>   <i>C++17, CUDA, Make</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | <b>2025</b> |
| <ul style="list-style-type: none"> <li>Wrote a Monte Carlo path tracer twice with zero external dependencies: a single-file C++17 CPU reference (654 lines) and a CUDA port (586 lines) mapping one pixel to one GPU thread in 16x16 blocks.</li> <li>Ported the renderer to the GPU by turning the recursive bounce path into a bounded iterative loop accumulating attenuation, with device memory managed via cudaMalloc/cudaMemcpy/cudaFree.</li> <li>Implemented Lambertian, metal, and dielectric materials from the math (Snell refraction, Schlick Fresnel), jittered supersampling for anti-aliasing, and a thin-lens camera for real depth-of-field.</li> <li>Made renders reproducible with a -seed flag: Mersenne Twister on the CPU, inline per-thread XORShift on the GPU to avoid global-state contention.</li> </ul> |             |
| <b>Kegel Trainer: Performance-Focused Flutter Wellness App</b>   <i>Flutter, Dart, Riverpod (AI-assisted)</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | <b>2026</b> |
| <ul style="list-style-type: none"> <li>Shipped a ~7,900-line Flutter app across 7 screens with zero bundled image assets; all art renders as inline SVG tinted from the active theme, scaling crisply at any size.</li> <li>Held frame rate stable on low-end phones by stripping blur and gradients from the render path, and cut the release APK from ~56 MB to 15-20 MB per ABI via R8 minification and ABI splitting.</li> <li>Built a bring-your-own-key AI coach over nine LLM providers that prompts from live app state, with user keys sent directly to the provider and never stored off-device.</li> </ul>                                                                                                                                                                                                                |             |

Education

|                                                                                                     |                           |
|-----------------------------------------------------------------------------------------------------|---------------------------|
| <b>Deenbandhu Chhotu Ram University of Science and Technology</b>                                   | <b>Sonipat, Haryana</b>   |
| <i>Bachelor of Technology in Electronics &amp; Communication Engineering (AI/ML Specialization)</i> | <i>Nov 2023 - Present</i> |

Work Experience

|                                                                                                                                                                                                                                                                                                                   |                               |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|
| <b>TH!NKBOTS, DCRUST, Murthal</b>                                                                                                                                                                                                                                                                                 | <b>October 2023 - Present</b> |
| <i>Robotics Club Coordinator</i>                                                                                                                                                                                                                                                                                  | <i>Murthal, Haryana</i>       |
| <ul style="list-style-type: none"> <li>Coordinate club activities and mentor junior members on robotics fundamentals: Arduino programming, sensor integration, and project development.</li> <li>Developed club projects including a PID-controlled Line Follower bot and a Multi-Terrain Gripper bot.</li> </ul> |                               |
| <b>E-Yantra Robotics Competition</b>                                                                                                                                                                                                                                                                              | <b>2024, 2025</b>             |
| <i>Team Leader</i>                                                                                                                                                                                                                                                                                                |                               |
| <ul style="list-style-type: none"> <li>2024: Led autonomous warehouse drone development using ROS 2, OpenCV, and Gazebo simulation.</li> <li>2025: Developed self-balancing lunar scout robot with PID/LQR control systems and Fusion 360 design.</li> </ul>                                                      |                               |
| <b>Smart India Hackathon</b>                                                                                                                                                                                                                                                                                      | <b>2024, 2025</b>             |
| <i>Team Leader and Event Organizer</i>                                                                                                                                                                                                                                                                            |                               |
| <ul style="list-style-type: none"> <li>2024: Team member; built a water conservation cross-platform game using Godot Engine.</li> <li>2025: Team leader and event organizer, alongside competing as a member.</li> </ul>                                                                                          |                               |
| <b>IEEE YESIST12 2026</b>                                                                                                                                                                                                                                                                                         | <b>2026</b>                   |
| <i>Finalist</i>                                                                                                                                                                                                                                                                                                   |                               |
| <ul style="list-style-type: none"> <li>Selected for the international final round in Indonesia with an AI-based cloud anomaly detection project.</li> </ul>                                                                                                                                                       |                               |
| <b>Hackathon Finalist: Jewellery E-Commerce Platform</b>                                                                                                                                                                                                                                                          | <b>2025</b>                   |
| <i>Finalist</i>                                                                                                                                                                                                                                                                                                   |                               |
| <ul style="list-style-type: none"> <li>Built a full stack e-commerce website for a jewellery business and reached the finalist round.</li> </ul>                                                                                                                                                                  |                               |