Jodh Singh

jetjodh@gmail.com

Education

- **Rutgers University** — New Brunswick, USA
  MS Computer Science — *Sept.2021–May 2023*
- **Guru Gobind Singh Indraprastha University** — India
  B.Tech Computer Science — *Aug.2016–June 2020*

Technical Skills

Languages: Python, C/C++, C/#, TypeScript, JavaScript
Frameworks: PyTorch, Svelte, React, Node.js, Flask, Django, FastAPI, PostgreSQL
Developer Tools: Git, Terraform, Docker, Kubernetes, CircleCI, OpenAI APIs, GCP, AWS, Azure, Linux

Experience

- **PictureStudio** — Remote
  Lead ML Engineer — *Aug 2023 – Present*
  - Collaborated on a end-to-end multimodal creation platform—React/Next.js + GraphQL frontend over Python micro-services—that lets artists **see, steer, and iterate** on diffusion & autoregressive models via spatial-control & prompt interfaces; now generates 1000+ images per day.
  - Designed and productionized a **LoRA/ControlNet training + inference** pipeline (PyTorch, ffmpeg, K8s, SLURM) that scales from a lone GPU to multi-node clusters; shaved P95 latency 30% and cut container build time 2× with layer caching.
  - Extended open-source frameworks (ComfyUI, A1111) with custom nodes for depth, style, and 3-D camera paths.
  - Launched self-serve fine-tuning UI—drag-and-drop asset upload, handling 1 – 10 000+ images/job while maintaining high subject/style fidelity via VLM auto-captioning and image pre-preprocessing.
  - Built a **100 k-prompt human + synthetic eval harness**; blocks quality regressions pre-deploy and feeds actionable metrics back to Research/Tech-Art for rapid model iteration.
  - Drove R&D on quantization and adapter training, unlocking on-device generation and new editing workflows now adopted by 90% of power users.
- **Thrasio** — NYC, USA
  Full Stack SWE Intern — *June 2022 - Sept 2022*
  - Ported existing RESTful APIs to GraphQL to improve reliability and scalability. Developed React components to interface with GraphQL for smooth data access. Leveraged infrastructure-as-code tools such as Terraform and Kubernetes for repeatable and automated environment provisioning.
  - Built a backend upload service with GraphQL served from S3 for resilient file processing. Implemented async multi-table data updates and Elasticsearch search for fast querying. Developed TypeScript hooks and pre-signed URL React components for streamlined S3 uploads.
  - Developed a barcode verification service using TypeScript, React, and PostgreSQL. Implemented custom GraphQL APIs for querying verification status. Hardened data validity checks and added monitoring for production robustness.
- **Shiryam Technologies** — India
  Backend Software Developer — *August 2020 - July 2021*
  - Architected modular CMS backend in Python/Flask, deployed on AWS for scalability. Reduced asset load times 5x to under 200ms via caching and async processing. Added Elasticsearch search with custom indexing for fast queries across files and DB.
  - Developed reusable automated data pipelines in Python for an algorithmic trading platform. Designed resilient DB ingestion mechanisms and failover handling. Created frameworks to add new trading algorithms with minimal code easily.
  - Integrated payment gateways and built out auth flows and hardened KYC processes for security.
- **BlueStacks** — India
  ML Intern — *June 2018 - Sept 2019*
  - Designed and implemented smart controls recognition system, a computer vision-based solution using PyTorch, integrated natively into the existing BlueStacks application stack. Built scalable ML infrastructure and CI/CD pipelines.
  - Achieved 98.78% accuracy and an average latency of 60 ms in real-world tests with a model size of just 10 MB.
  - Built an in-game OCR system using DL based solution using RCNNs, and various image processing techniques. Achieved 77% image-to-text accuracy in real-world English, French, German, Korean, Mandarin, and Japanese.

Projects

- **Production level inference sever**: Implemented an LLM-focused inference server in FastAPI with graph optimizations with ONNX runtime. Used IaaC tools like Terraform to deploy the server on AWS EC2 instance on K8s. Implemented Slurm scheduling policies for serving, monitoring, troubleshooting and enhancing cluster performance.
- **Short story generation using a single reference image**: I built an image synthesis system using CLIP embeddings from the reference images, extrapolating them to create a narrative. To create a coherent story, I used them to generate multiple images chronologically and logically related to previous reference images.
- **Hateful Meme Detection**: Facebook's Hateful Memes competition in which, using multi-modal techniques, the memes were classified as hateful or non-hateful. I used multiple data augmentation techniques to generate data to improve the accuracy of finetuned LLMs. The highest rank achieved was 1, finally in the top 50.