

Tejas Patel

✉ tejasbiharibhai.patel@slu.edu | [In](#)

<https://www.linkedin.com/in/tejas-patel-15816b19a/>

<https://github.com/t Patel9824>

AI/ML Engineer, AI Developer, ML Data Engineer — LLMs, NLP, Gen AI, MLOps

EDUCATION

M.S. in Computer Science (AI/ML focus)

Saint Louis University (Jan 2024 – Dec 2025)

Relevant Coursework: Machine Learning, Deep Learning, NLP, Computer Vision, Data Mining, AI Algorithms

Hands-on experience with LLM finetuning, NLP, and computer vision using PyTorch, Hugging Face, and Databricks. Built end-to-end projects around feedback generation, CLIP-based zero-shot image classification, and an AI-powered mental health support app.

TECHNICAL SKILLS

Languages Python, Java, SQL, JavaScript, Bash, C/C++

ML / DL PyTorch, TensorFlow, scikit-learn, PyTorch, Hugging Face Transformers/PEFT, CLIP, XGBoost, CatBoost

NLP Prompt engineering, RAG, tokenization, evaluation (BLEU/ROUGE), text classification, summarization

Vision OpenCV, zero-shot classification (CLIP), data augmentation, transfer learning

MLOps MLflow, Weights & Biases, model versioning, inference APIs (FastAPI), Docker, experiment tracking

Data pandas, NumPy, Matplotlib, Databricks (PySpark), Delta Lake, data validation

Cloud AWS (S3, EC2 basics), Git/GitHub, Linux; basic Kubernetes & containerization

Optimization LoRA/QLoRA, quantization (int8/4), Unsloth, ONNX export (basic)

SELECTED AI/ML PROJECTS

Generative AI for Automated Feedback in CS Courses

LLMs, NLP — 2025

Collected and curated a dataset of 1,000+ CS question-answer pairs and trained a baseline text classifier (Python, scikit-learn) to predict rubric-based labels. Prompt-engineered an LLM (LLaMA / GPT-style) to generate rubric-aligned feedback for student answers and evaluated quality using accuracy/F1 and targeted human review.

CLIP Zero-Shot Image Classification (Flowers-102)

Computer Vision — 2025

Applied CLIP for zero-shot classification on the Flowers-102 dataset and analyzed per-class accuracy and confusion matrices. Tested different prompt templates and simple ensembling to study how text descriptions affect model performance in zero-shot settings.

Fast LLM Finetuning with Unsloth + LLaMA 3.2

Instruction Tuning — 2025

Finetuned an instruction-tuned LLaMA 3.2 model using Unsloth with LoRA/QLoRA to run under limited GPU memory. Compared training speed, memory usage, and response quality across adapter sizes and hyperparameters to understand efficiency–performance trade-offs.

Mindheaven — Mental Health Support (NLP Prototype)

Product/NLP — 2025

a mental health support web app with blogs, mini-games, guided exercises, music playlists, therapist booking, and an AI chatbot using Ollama. Implementing backend and AI components using [your real stack, e.g., Python/FastAPI or Node.js, LLM APIs, simple RAG] and a responsive web UI with focus on safe, empathetic responses

a complete EEG decoding system using PhysioNet data, including preprocessing, feature engineering, and model training. Implemented classical ML (SVM) and deep learning (EEGNet) architectures for motor-imagery classification. Evaluated model performance with training curves and confusion matrices, demonstrating practical BCI signal-processing skills.

ADDITIONAL

Interests: Efficient finetuning, evaluation design, human-in-the-loop feedback, Brain Computer Interface research, Signal decoding.

Technical Keywords: Machine Learning, Deep Learning, Generative AI, LLMs, NLP, Text Classification, Prompt Engineering, CLIP, Zero-shot Image Classification, Computer Vision, Python, scikit-learn, LLaMA 3.2, Unslot, LoRA, QLoRA, Instruction Tuning, Parameter-efficient Finetuning, Model Evaluation, Dataset Creation, AI Chatbot, Mental Health Support Web App, RAG, FastAPI/Node.js, Web APIs.