

SHUBHAM GAJJAR

Portland, Maine | (207) 332-2039 | gajjar.shu@northeastern.edu | linkedin.com/in/implici7 | shubhamgajjar.dev

EDUCATION

Northeastern University , Portland, Maine <i>Master of Science in Artificial Intelligence</i>	September 2025 – May 2027
LDRP Institute of Technology and Research , Gandhinagar, India <i>Bachelor of Engineering in Computer Engineering</i> , Grade Point Average: 8.41/10.0	September 2022 – May 2025
VPMP Polytechnic , Gandhinagar, India <i>Diploma in Computer Engineering</i> , Grade Point Average: 9.22/10.0	September 2019 – May 2022

Relevant Coursework: Machine Learning, Deep Learning, Computer Vision, Data Structures and Algorithms, Image Processing

TECHNICAL SKILLS & CERTIFICATION

Programming Languages: Python, JavaScript

Deep Learning: TensorFlow, Keras, PyTorch, CUDA

Computer Vision: OpenCV, Matplotlib, Albumentations

Data Science: NumPy, Pandas, Jupyter, Scikit-learn

Web Development: Next.js, React, JavaScript, Tailwind CSS, Flask

Tools: Git, Vercel, Framer Motion

Certificates: Python for Data Science from Indian Institute of Technology Madras, Python Data Structures from University of Michigan

PROFESSIONAL EXPERIENCE

BigCircle (UPSAAS Technologies LLP), Gandhinagar, India

Artificial Intelligence Engineer

January 2025 – August 2025

- Developed end-to-end Deep Research pipeline using Python orchestrating ML prompt generation, Firecrawl API for web scraping, ChatGPT API summarization, graph visualization with Matplotlib and Seaborn, and automated Typst PDF reports; optimized API call efficiency reducing processing time by 75%
- Engineered pagination and authentication systems using JavaScript and Next.js, accelerating page load times by 40% with Docker containerization for 500+ concurrent sessions
- Collaborated with 5-member Agile team using Git for version control; performed code reviews to improve code quality

KEY PROJECTS & PUBLICATIONS

Hybrid ResNet-ViT for Skin Cancer Classification

Published in IEEE, 2025

- Hybrid architecture using TensorFlow combining frozen ResNet50 with four-head Vision Transformer blocks, achieving 96.3% accuracy and macro F1 of 0.961 on HAM10000 dataset with 10,015 images preprocessed using Albumentations
- Integrated Global Average Pooling and multi-head self-attention for seven-class classification with NumPy arrays, achieving Area Under Curve of 1.00; visualized performance using Matplotlib
- Presented research at IEEE World Conference on Applied Intelligence and Informatics to 100+ attendees

Extended ResNet50 with Inverse Soft Mask Attention for Skin Cancer

Submitted to journal, 2025

- Two-stage pipeline using TensorFlow and Keras combining U-Net++ hair segmentation with Extended ResNet50 classifier featuring Inverse Soft Mask Attention, achieving 97.89% accuracy on HAM10000 dataset with 10,015 images preprocessed using OpenCV.
- Integrated dense residual blocks and Squeeze-and-Excitation modules with learnable weighted feature aggregation for occluded and unoccluded regions
- Nadam optimizer with Cosine Decay Restarts; conducted 21 architectural trials with custom attention mechanisms outperforming SCCNet (95.20%) and SPCB-Net (97.10%)

VGG16-MCA UNet for Brain Tumor Segmentation

Under Review at Elsevier

- Led innovation by designing a VGG16-based encoder with a multi-channel attention decoder using TensorFlow, achieving 99.59% accuracy and 99.71% specificity on the LGG Brain MRI Segmentation dataset from 110 low-grade glioma patients
- Implemented ensemble learning combining multiple model configurations in Keras, improving the Dice coefficient by 3.7% over standard UNet through analytical thinking and systematic hyperparameter tuning
- Used data engineering to create a preprocessing pipeline with NumPy and Pandas for data handling, implementing skull stripping, intensity normalization using Scikit-learn, and resizing FLAIR MRI scans to 256x256 pixels, following ethical guidelines for AI in medicine

Reinforcement Learning Agent for TrackMania

Academic Project, 2024

- Autonomous racing agent using PyTorch implementing Implicit Quantile Networks with NumPy for state processing, achieving 85% track completion through iterative reward tuning
- Optimized training framework using CUDA for GPU acceleration, reducing training time by 60% through parallel environment simulations