SHUBHAM GAJJAR

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

EDUCATION

Northeastern University, Portland, Maine — September 2025 – May 2027
Master of Science in Artificial Intelligence

LDRP Institute of Technology and Research, Gandhinagar, India — September 2022 – May 2025
Bachelor of Engineering in Computer Engineering, Grade Point Average: 8.41/10.0

VPMP Polytechnic, Gandhinagar, India — September 2019 – May 2022
Diploma in Computer Engineering, Grade Point Average: 9.22/10.0

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

TECHNICAL SKILLS & CERTIFICATION

Programming Languages: Python, JavaScript, TypeScript, Next.js, React
Deep Learning: TensorFlow, Keras, PyTorch, CUDA
Computer Vision: OpenCV, Matplotlib, Albumentations
Data Science: NumPy, Pandas, Jupyter, Scikit-learn
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

- Architected multi-agent API system using Python and Flask with distributed computing, reducing report generation from 20 to 5 minutes for 10,000+ queries
- Engineered pagination and authentication systems using JavaScript and Next.js, accelerating page load times by 80% with Docker containerization for 500+ concurrent sessions
- Delivered iOS applications using React Native, increasing mobile engagement by 45% within first quarter
- Collaborated with 5-member Agile team using Git for version control; performed code reviews improving quality metrics by 30%

KEY PROJECTS & PUBLICATIONS

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 and Albumentations
- 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%)

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

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 and Keras, achieving 99.59% accuracy and 99.71% specificity on the LGG Brain MRI Segmentation dataset from 110 low-grade glioma patients processed using OpenCV for image manipulation
- 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