Mukund Telukunta

https://mukundtelukunta.com

(952) 649-2653
mukundtelukunta@gmail.com
https://www.linkedin.com/in/mukundtelukunta

Summary

Graduated with PhD in Computer Science (May 2025) with 5+ years of experience building and deploying machine learning systems across NLP, computer vision, graph learning, and audio AI. Designed production-grade pipelines (WaveCrafter), pre-trained large Transformer and GNN-based models (DNABART), and implemented state-of-the-art architectures including Vision Transformers and Graph Convolutional Networks from research papers. Published researcher (CHIL 2024, ECML PKDD 2023) with a strong record of translating complex ML research into robust, maintainable software.

Education

Missouri University of Science and Technology — Rolla, MO
PhD in Computer Science — May 2025
Research Areas: Algorithmic Fairness, Computer Vision

Technical Skills

Languages: Python, Java, C, C++, R, JavaScript, HTML/CSS
Machine Learning & Deep Learning: PyTorch, TensorFlow, Keras, HuggingFace Transformers, scikit-learn, NumPy, SciPy, Pandas, Weights & Biases (W&B)
Model Architectures: CNN, RNN/LSTM, Vision Transformers (ViT), Graph Convolutional Networks (GCN), BART (encoder-decoder), YOLOv6, ResNet
Computer Vision: OpenCV, object detection, image segmentation, Grad-CAM, Kalman Filtering
Cloud & Deployment: AWS, GCP, Heroku, Modal.com, GitHub Actions (CI/CD), Docker

Selected Publications (Google Scholar)

1. **Telukunta, M.**, Stuart, M., Nadendla, V. S. S., & Canfield, C. "Fairness Perception of Regression-based Predictive Analytics Tools used in Kidney Placement." Under Review. *Preprint*
2. **Telukunta, M.**, Rao, S., Stickney, G., Nadendla, V. S. S., & Canfield, C. (2024). "Learning Social Fairness Preferences from Non-Expert Stakeholder Opinions in Kidney Placement." *CHIL 2024*.
3. **Telukunta, M.**, & Nadendla, V. S. S. (2023). "Towards Inclusive Fairness Evaluation via Eliciting Disagreement Feedback from Non-Expert Stakeholders." *Bias and Fairness in AI Workshop, ECML PKDD*.

Experience

Missouri University of Science and Technology — Rolla, MO
ML Research Scientist — August 2020 - May 2025

- Implemented 10+ state-of-the-art ML/DL architectures from research papers – including Vision Transformers, Graph Convolutional Networks, and encoder-decoder Transformers – adapting them to novel problem settings with accuracy improvements of up to 24%.
- Designed and conducted 15+ controlled experiments (randomized trials, within-subjects designs) to evaluate ML model behavior; performed statistical analysis using Python and R including ANOVA, regression, propensity score matching, and mediation analysis.
- Built automated data pipelines and reproducible ML workflows using Python and GitHub, reducing experiment iteration time by 60% across multiple concurrent projects.
- Designed and deployed a large-scale survey experiment on Prolific (N=5,000+ responses) using Power Analysis to determine sample sizes, achieving statistical power of 0.85+.
- Mentored 2 undergraduate researchers on ML methodologies, experimental design, and software engineering best practices, enabling independent execution of model training and evaluation workflows.
- Published 5 peer-reviewed papers at top ML/AI conferences and workshops (CHIL 2024, ECML PKDD 2023, RecSys 2020); collaborated with cross-functional teams including clinicians, engineers, and policy stakeholders.

AT&T Labs Des Peres, MO
Student Technical Intern June - August 2019

- Architected a cloud-based reporting system using PHP, JavaScript, HTML, AWS, and GCP, delivering real-time financial performance dashboards that improved forecasting accuracy by 30% across an 800-person organization.
- Engineered automated ETL pipelines connecting large-scale relational (RDBMS) and NoSQL databases, reducing report generation time by 85% and saving approximately 180 working hours annually.
- Deployed automation solutions achieving 95% user adoption within 6 months by collaborating with stakeholders to translate business requirements into reliable technical implementations.

Projects (https://github.com/mukund0911)

PitchVision — ViT-Based Automated Football Understanding System

- Implemented **ResNet-18 + YOLO** detector from scratch in pure **PyTorch** (BasicBlock skip connections, anchor-based grid decoding, greedy NMS with IoU thresholding); achieved **87.3% mAP0.5** on player/ball detection at **42 FPS on RTX 4090** (416×416 input).
- Built **DeepSORT multi-object tracker** from first principles - 8D Kalman filter (constant-velocity motion model), 128-dim L2-normalized ReID CNN embeddings, cascade matching via Hungarian algorithm on cosine + IoU costs; maintained **MOTA 71.4 / IDF1 68.9 with <3% ID switches** across 90-min matches.
- Designed a **2M-parameter tactical Transformer** (6-layer pre-norm encoder, 8-head attention, Fourier positional encodings for continuous 2D coordinates) predicting off-ball intent across 5 classes, 8-way run direction, urgency, and Δ-position; **reached 78.6% intent accuracy and 1.4m MAE** on next-position prediction over SoccerNet-v3 tracking data.
- Engineered an end-to-end video pipeline (homography calibration via cv2.findHomography RANSAC, pixel → field projection, heuristic event detection for passes/shots/dribbles/presses) producing per-match scouting reports at <100ms/frame CPU, <20ms GPU.
- Shipped attention-rollout interpretability (Abnar & Zuidema 2020) surfacing ball→player causal influence, and exposed the event log through a FastAPI query layer supporting both structured stats endpoints and LLM-routed natural-language scouting questions.

Graph Convolutional Network (GCN) for Urban Mobility (Stealth Startup)

- Built a **Heterogeneous Graph Transformer (HGT)** and **GCN**-based system operating over a graph of **700K nodes and 1.8M edges**, modeling complex relational structures for urban mobility prediction and graph-structured inference tasks.
- Designed custom graph data loaders and mini-batch sampling strategies using **PyTorch Geometric (PyG)** to efficiently handle large-scale heterogeneous graphs without memory overflow; reduced per-epoch training time by 35% through neighbor sampling optimization.
- Implemented multiple GNN variants (GCN, GraphSAGE, GAT, HGT) in a unified framework, enabling controlled comparison of message-passing architectures on downstream node classification and link prediction tasks.
- Applied graph-based knowledge representation to the DNABART project, using GNNs to mine relationships in genomic knowledge graphs and augment sequence model predictions with structured biological context.

WaveCrafter: AI-Powered Multi-Speaker Voice Editor | *Try it:* https://wave-crafter.com/

- Built a full-stack audio editing platform using **React**, **Flask**, **PyTorch**, and **VoiceCraft (330M parameters)** for AI voice cloning. Enables text-based speech modification with **95%+ voice similarity** across multi-speaker audio files using zero-shot learning and automatic speaker diarization.
- Implemented **MFA (Montreal Forced Aligner)** for word-level precision editing, improving **voice cloning quality from 60% to 95%+** similarity for insertions. Developed dual-mode inference architecture using Edit Mode for text modifications and TTS Mode for full audio generation with word-boundary alignment.
- Deployed serverless GPU pipeline on **Modal.com** with **A10G GPUs** and auto-scaling up to 10 containers. Optimized voice cloning latency to **6-8 seconds per segment** (down from 15-18s cold

start) through container warmup (600s) and parallel processing (3x speedup).
- Developed intelligent reference audio selection using composite quality scoring across 6 metrics including **SNR**, **RMS energy**, and **spectral richness**. Built custom speech/music classifier using **scikit-learn** with MFCC feature extraction processing **30-minute podcasts in ∼46 seconds**.
- Architected cloud solution using **AWS S3**, **Heroku**, **Modal serverless GPU**, and **AssemblyAI API**, reducing costs by 90% ($350 to $33/month). Implemented **GitHub Actions CI/CD** for automated deployment to GitHub Pages and Heroku.
- MD5-based caching achieving ¡1s response for repeated uploads, and integrated OpenAI GPT-3.5 for dialogue generation.

DNABART: A Genomic LLM Foundational Model for Sequence Correction and Classification
- Developed a large language model for **genomic sequence error correction** and classification using **BART encoder-decoder** architecture with Graph Neural Networks (GNN) for knowledge graph mining. Built with PyTorch and tracked experiments using **Weights & Biases**.
- Implemented **Byte-Pair Encoding (BPE)** tokenization for genomic sequences, capturing long-range dependencies in large-scale DNA datasets. Created data collection frameworks for structured and unstructured genomic data processing.
- **Pretrained** BART on DNABERT2 (15GB) and Saccharomyces Genome Database (1M sequences) using synthetic sequence corruption strategies, leveraging AdamW optimizer and a linear learning rate scheduler for efficient model convergence on NVIDIA H100 and V100 GPUs.
- **Fine-tuned** BART on the Genome Understanding and Evaluation (GUE) benchmark, achieving state-of-the-art results on 16 out of 26 tasks, including epigenetic mark prediction and core promoter detection, with accuracy improvements of up to 24%.
- Engineered scalable training pipelines for sequence error correction and classification, optimizing compute efficiency on the Missouri S&T HPC cluster.

Real-time Multi-Sensor Multi-Object Detection via Transfer Learning
- Fine-tuned the YOLOv6 object detection model using transfer learning to detect NVIDIA Jetson Robots on a custom-built testbed, achieving a detection accuracy of 98.5%.
- Designed and developed a real-time multi-sensor vision system by integrating images from multiple camera sensors. Leveraged OpenCV Stitcher class to create seamless panoramic views and implemented advanced detection algorithms like SIFT (Scale-Invariant Feature Transform) for feature extraction.
- Implemented Kalman Filtering for robust real-time tracking of Jetbot robots, including state vector estimation and covariance matrix updates. Achieved a tracking accuracy of 95% under dynamic conditions, with precise object state prediction and noise filtering.
- Engineered a multi-camera localization system by positioning multiple sensors as satellites around the testbed to triangulate and determine robot positions. This ongoing effort aims to improve localization accuracy by 20%, enabling sub-centimeter precision for real-time robot tracking.
- Optimized system pipeline for real-time performance using OpenCV, Kalman Filtering, and YOLOv6 on NVIDIA Jetson hardware, ensuring low-latency operation and computational efficiency.