

# JACOB LANE MOORE

Huntsville, AL 35806 | jlmoors001@gmail.com | (256) 701-7888

## Education

**MS, Georgia Institute of Technology**, Computer Science, Computing Systems Expected 2027

- GPA: 4.0/4.0
- **Coursework:** Hypervisors, Distributed Systems, Software Architecture, Compilers

**BS, Mississippi State University**, Computer Engineering Graduated

- GPA: 4.0/4.0
- **Coursework:** Embedded Systems, Systems Programming, Computer Architecture, Operating Systems

## About Me

- **Languages:** C++/C, Python
- **Development:** Neovim, VSCode, Linux, Docker, CMake, Make, Zsh/Bash, Git
- **ML/DL:** PyTorch, Tensorflow, RLLib, Scikit-Learn, Pandas, Grafana, Matplotlib
- **Certs:** DeepLearning.AI, Neural Networks and Deep Learning (Andrew Ng), CompTIA Security+

## Experience

**Software Design Engineer**, Torch Technologies – Huntsville, AL 05/2023 – Current

- Support a core Reinforcement Learning project, developing libraries and functionalities critical to the training/integration pipeline.
- Develop and maintain a C++ data model library for manipulating datasets used in deep learning architectures (Transformers, CNNs).
- Build C++ applications for AI classification performance metrics and ONNX model inference integration.
- Developed a Python AI classification pipeline simplifying data generation, preprocessing, training, and evaluation for ML/DL models.
- Utilize Python multiprocessing to efficiently handle large datasets, optimizing memory usage and runtime.
- Collaborate in an AGILE environment, managing code via GitLab and ensuring reliable AI systems.
- Research state-of-the-art DL and RL architectures to improve workflow and performance.

**Software Engineering Intern**, Torch Technologies – Huntsville, AL 05/2020 – 08/2022  
(Summers/Winters)

- Supported and added features to a primary C++ project with 12+ distinct services.
- Briefly supported an Unreal Engine project, expanding socket communications for enhanced real-time interaction.
- Created a Python GUI for viewing/zooming Digital Elevation Models and overlaying satellite imagery.

**Undergraduate Research Assistant**, Mississippi State University CSE Department 09/2020 – 05/2022

- Developed data acquisition/analysis systems for predictive maintenance using a machine fault simulator and fiber optic sensors.
- Researched transfer learning and tested advanced ML clustering algorithms (TICC).
- Built and studied GANs for high-quality dataset expansion.

## Skills

---

- Skilled in C++/Python implementation of design patterns and custom data structures.
- Strong experience with CMake-based build/install processes on Windows/Linux.
- Familiar with IPCs and synchronization primitives (ROS2 development).
- Extensive Python/Bash scripting for automation and environment setup.
- Proficient in manipulating large datasets with Pandas/NumPy.
- Strong understanding of deep learning (LSTM, RNN, Transformers, TFT).
- Experience with ML algorithms (GBM, KNN, K-Means, Random Forest).
- Experience architecting medium-scale OO applications/libraries in C++ and Python.
- Proficient in Linux system administration and development.
- Skilled in collaborative development, coding standards, and version control (Git).

## Projects

---

### Autonomous RC Car (Minesweeper):

- Solely engineered the entire software stack for a low-cost, disposable autonomous RC car designed to sweep for and detonate landmines.
- Architected a robust ROS 2 pub/sub service-oriented navigation system fusing raw IMU/GPS data for precise Ackermann steering control via electronic speed controllers.
- Implemented a Python-based command controller integrating base station input and RTK correctional data over a mavros radio, ensuring accurate real-time navigation.
- Developed all supporting base station software and communication interfaces, seamlessly linking computational logic with field operations.

### Tiger Language Compiler:

- Designed and implemented a C++ compiler for a functional, ALGOL-like language (Tiger), producing optimized MIPS assembly.
- Introduced a TAC intermediate representation and implemented two register allocation strategies:
  - Chaitin's Naive Allocation: A simpler, less optimized approach relying on a heuristic-based allocation.
  - Briggs Optimization: A comprehensive scheme incorporating control-flow and liveness analysis, as well as instruction cost metrics to minimize load/stores and branches.
- Achieved best-in-class performance on benchmark tasks, reducing instruction counts by a factor of 10 compared to peer solutions.

### Before and After Creator:

- Built a locally hosted web application leveraging unsupervised ML and deep learning to match before/after car detailing images.
- Automated bulk image processing, generating side-by-side collages for a detailing business, improving workflow efficiency and consistency.
- Engineered a custom pairing algorithm to reliably identify corresponding images and streamline final outputs for user download.