

# Siddharth Singh Solanki

+1-929-592-1785 • [solankisiddharthsingh@gmail.com](mailto:solankisiddharthsingh@gmail.com) • [SiddSS.github.io](https://github.com/SiddSS) • [LinkedIn](#)

## Education

- Georgia Institute of Technology**, Atlanta, USA [2022 - 2024]
  - Master's in Computer Science (Specialisation: Machine Learning) GPA 3.92/4.00
- Indian Institute of Technology Goa**, Farmagudi, India [2018 - 2022]
  - B.Tech in Computer Science and Engineering (**CPI 9.72/10**) Bronze Medalist, ranked second in the batch

## Technical Skills

- **Programming:** Java, Python, C++, JavaScript
- **Software/Libraries/Misc:** PyTorch, MATLAB, ReactJS, Spring framework, Bash, OpenGL, PostgreSQL

## Experience

- **Wayfair**, Boston, USA [Jun'24 - Present]  
*Software Developer I*
  - Working with Java spring microservices in the Warehouse Receiving team.
  - Wrote Kafka listeners and producers, optimized database transactions. **Resolved time critical bugs** impacting warehouse operations.
  - Played a key role in productionizing and developing a slackbot that uses RAG **uses past issue tickets data to predict solutions** and help reduce the down time.
- **MathWorks**, Natick, USA [May - Aug '23]  
*Software Development Intern*
  - Worked with MATLAB and C++ codebase along with Simulink's parallel compute library.
  - Developed a function that allows users to define and execute custom progress trackers and plots for parallelized simulations without compromising the simulation speed.
  - It **reduces the execution time for a typical user workflow upto 10X** for simulations involving 3-D plots in aerospace and robotics applications.
  - Wrote unit, system tests and customer facing documentation. **Code was shipped with 2024-a release of MATLAB.**
- **MathWorks** Hyderabad, India [Jun - Nov '21]  
*Software Development Intern*
  - Worked with C++, MATLAB and JavaScript codebases.
  - Optimized automated CNN deployment feature for Intel architecture GPUs. Performed memory optimizations through techniques such as buffer reuse, layer fusion and minimized data transfer between CPU and GPU.
  - **Achieved 2X speedup in training popular CNNs** such as ResNet, VGG-16 and AlexNet.
  - Performed testing for deployment, code was shipped with 2022-b, 2023-a release of MATLAB.
- **Machine Vision Lab - IIT Roorkee**, Roorkee, India [May - Aug '20]  
*Research Intern*
  - Computer Vision research under Prof. Balasubramanian Raman. Worked on moving hand sign recognition problem.
  - Used optical flow for sampling frames and extracted features by detecting keypoints and finetuning Resnet-50 and Inception-V3 on sign language datasets.
  - **Developed a novel Recurrent Neural Network architecture** for sequence learning and **implemented an optimized version** of the developed proof of concept using PyTorch.

## Projects

- **Data Augmentation using diffusion models** — [Summary Video](#) [2023]
  - Used diffusion models to substitute image augmentations in the contrastive learning approach used in the paper [SimCLR](#).
  - Increased Top-1 accuracy by 9 percent on the imagenet dataset along with better compute efficiency on training.
- **Stay Alive Think and Drive App** — [GitHub](#) [2023]
  - A web application which helps users to plan their journey by providing safety features based on past accident data, and live current weather conditions on the route.
  - The app has a React frontend and Mongo DB backend. Integrated with google maps API and weather APIs that work live with geolocation after the user inputs a travel route.
- **Reliable Answer Deduction** — [Project page](#) [2022]
  - Fine tuned BERT based LLMs and experimented with different attention mechanisms to develop a model which gives answers to the questions asked from a given comprehension.
- **Trash Classification** — [GitHub](#) [2020]
  - Cleaned and augmented TACO trash dataset. Modified the convolutional layers of a lightweight SSD7 object detector.
  - The developed model is edge deployable and can identify and classify upto 7 different trash categories and outputs bounding boxes over all the instances of trash in an image.