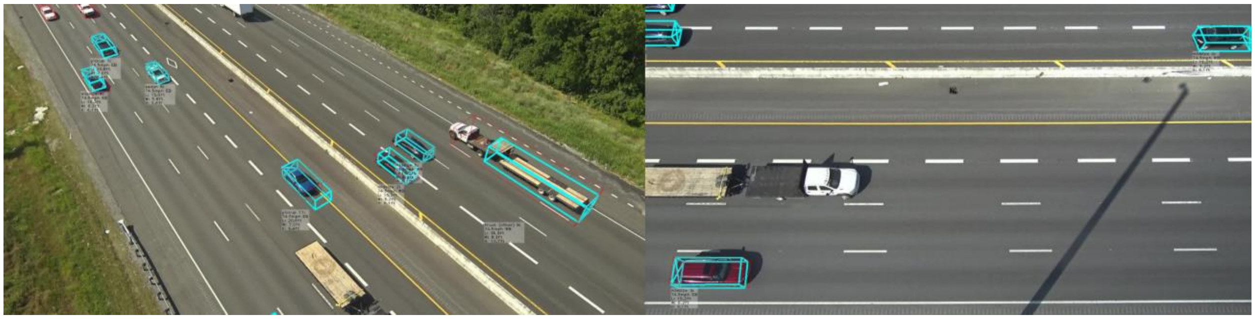
Vanderbilt Undergraduate Research Opportunity:

Computer Vision Research



We are seeking an undergraduate research assistant to conduct research tasks related to computer vision, object detection, and neural network training. Our team works heavily with algorithms for vehicle detection, tracking, and post-processing trajectory data. Existing object detection formulations are ill-posed for our task, so we seek to implement new loss functions and network architectures to better detect vehicles in the domain of overhead traffic camera video. The first research task will be the implementation and benchmarking of a novel loss function better suited to this domain.

The ideal candidate is an advanced undergraduate student who is eager to conduct research during spring 2022 with the possibility for extension into the summer and fall 2022. Candidates should have the following qualifications:

- Proficiency in Python and/or C++
- Proficiency with efficient matrix/tensor operations
- Strong familiarity with Pytorch (preferred) or TensorFlow
- Ability to work on problems of moderate complexity with limited oversight
- Availability - Spring 2022: 10-20 hours per week; possibility of summer 2022 at 20-40 hours per week

This project is conducted by Derek Gloudemans under the supervision of Professor Dan Work. Interested applicants should email derek.gloudemans@vanderbilt.edu a copy of their resume, along with answers to the following questions:

1. Consider a set $target$ of 2D rectangular bounding boxes parameterized by 4 coordinates $[xmin, ymin, xmax, ymax]$. Express the set as a tensor of dimension $[n\_boxes, 4]$. In Pytorch or TensorFlow, write a differentiable expression for the Intersection-over-Union (IOU) overlap computed row-wise between $target$ and $output$, a tensor of the same size.
2. In Pytorch or TensorFlow, write an optimization routine using stochastic gradient descent or similar to modify the parameters of $output$ to maximize the IOU metric between $target$ and $output$.
3. Consider two sets of 2D rectangular bounding boxes $boxes\_a$ and $boxes\_b$, expressed as tensors of sizes $[a, 4]$ and $[b, 4]$, respectively. Calculate the IOU between each pair of boxes selected from $boxes\_a$ and $boxes\_b$ respectively. Thus, the output should have size $[a, b]$. The function should not use loops (i.e. use tensor/matrix operations such as view(), reshape() and repeat() instead.)
4. What are limitations of 2D bounding box representations? Specifically, suggest a superior object representation for vehicles that more fully captures relevant information about the vehicle.