DeepROAD: A Multifaceted Deep Learning Suite for Real-Time Optimized Autonomous Driving

Edwin Weill, Jesse Tetreault, Varun Praveen, Dr. Melissa Smith
Future Computing Technologies Laboratory
Department of Electrical and Computer Engineering
College of Engineering, Computing, and Applied Sciences, Clemson University

Project Overview/Goals

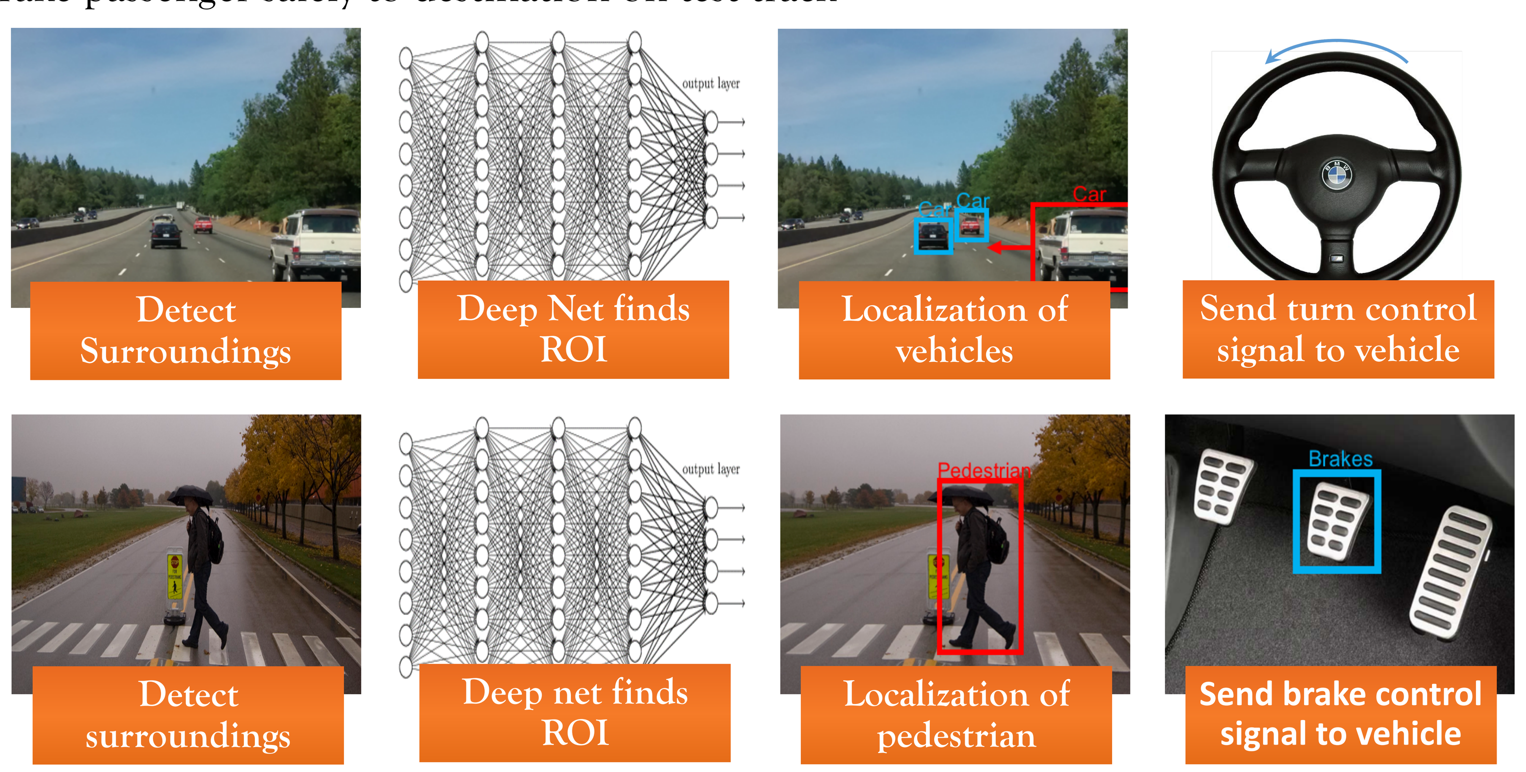
Project Overview:

Deep learning is a large research area that requires substantial amounts of computation as well as domain knowledge to optimally solve a given problem. CU-ICAR[1] has tasked our group with the development of a **perception** system for ADAS applications that has the ability to sense its surroundings to a degree of certainty that allows for the precise maneuverability of the vehicle around a self-designed test track safely while meeting certain real-time constraints. Our specific focus is to be able to perform as many useful tasks as possible that aid in the safe operation of a vehicle while still operating in real-time on efficient and practical acceleration hardware.

The hardware that utilized for these experiments is described below in the "Hardware/Software & HPC" section. All training is completed either on a machine with a single GPU (K20, Titan X) or on the Palmetto Cluster[2] which houses a many K40 GPUs that can be used for training. A K80 will be used in future training for performance comparisons. For this particular project, the Jetson TX1 will be used for inference as the POC on an embedded platform.

Project Goals:

- Remain in lane while vehicle in motion (no drifting)
- Follow all traffic laws (stop at STOP signs, etc.)
- Park in specified parking spot and pick up passenger
- Take passenger safely to destination on test track



[1]Clemson University International Center for Automotive Research
[2]https://palmetto.clemson.edu (Clemson's Palmetto Cluster)

Hardware/Software & HPC

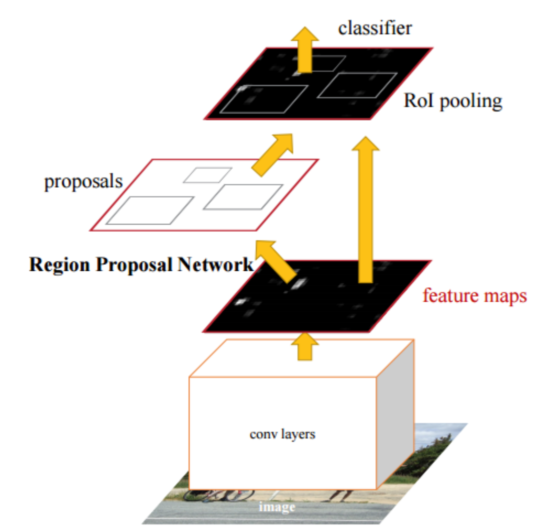
There are many hardware and software platform options that enable deep learning techniques to be accurate and expeditious. The following is the set of hardware and software utilized for this project for both training of deep learning networks and inference.

Hardware[1]	Software
NVIDIA K20*, K40*, Titan X*, K80*	Caffe, cuDNN,
NVIDIA Jetson TX1**	RCNN, Faster-RCNN, SegNet, TensorRT

*Training **Inference

Deep Learning

Detection

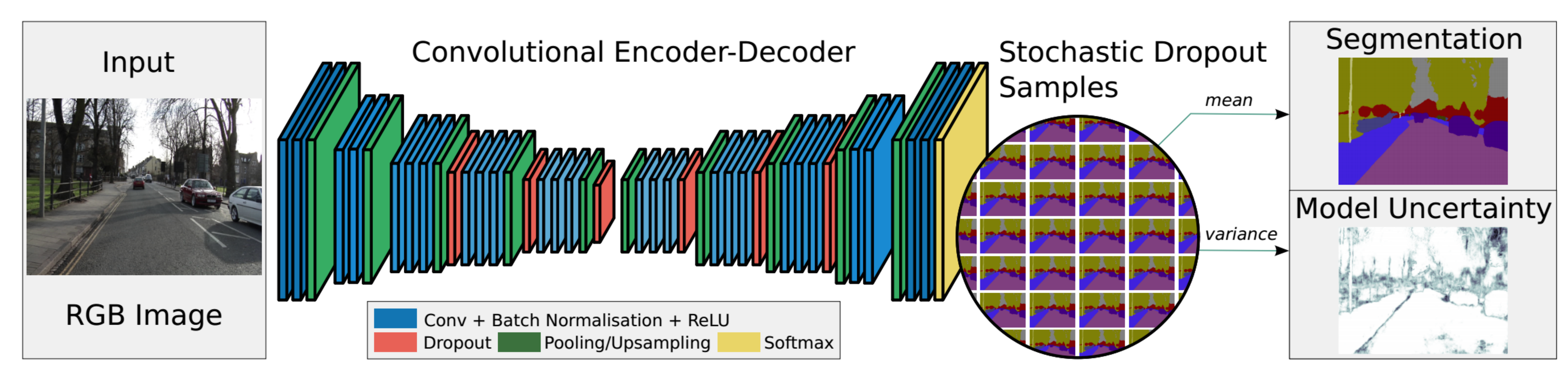


- Convolutional layers produce feature map
- Region proposal network localizes regions of interest
- Final layer combines feature map and region proposals for decision making
- Provides bounding box and probability values for each region over threshold
- Faster R-CNN and NVIDIA's TensorRT and DetectNet for fast inference

Ren, Shaoqing, et al. "Faster R-CNN: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems*. 2015.

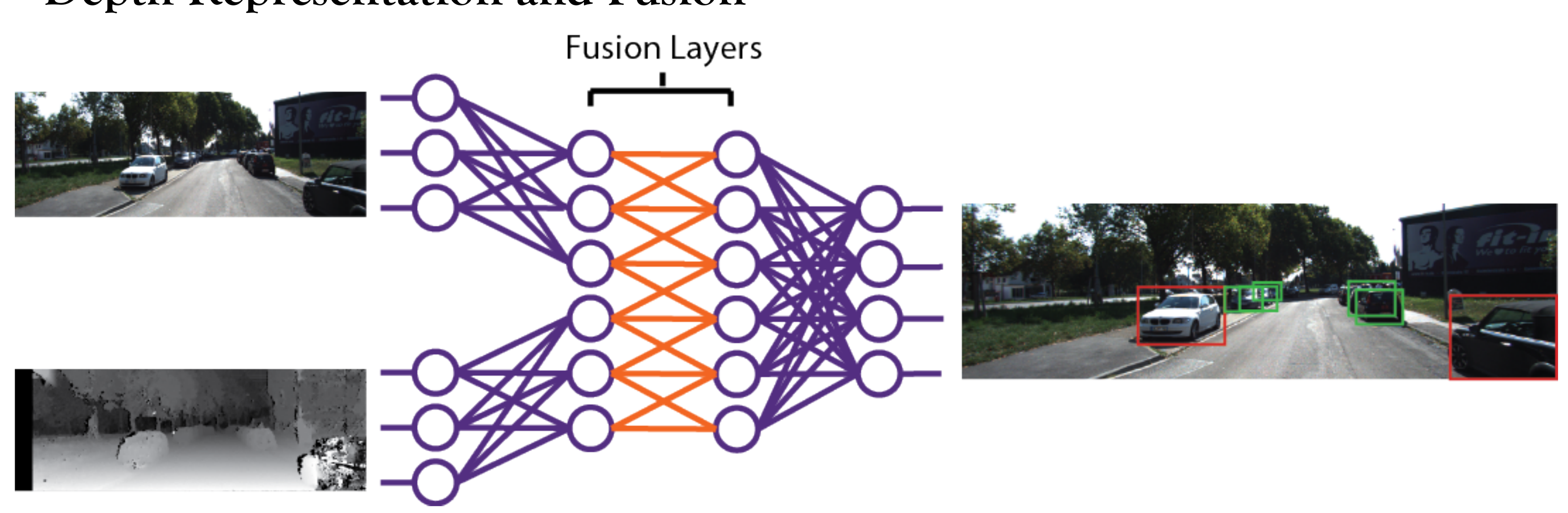
Segmentation

- Pixel-wise semantic labeling (trained on CamVid and Cityscapes)
- Convolutional Layers followed by Deconvolutional Layers
  - Deconv layers replace fully connected layers from VGG16



Badrinarayanan, Vijay, Ankur Handa, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling." *arXiv preprint arXiv:1505.07293* (2015).

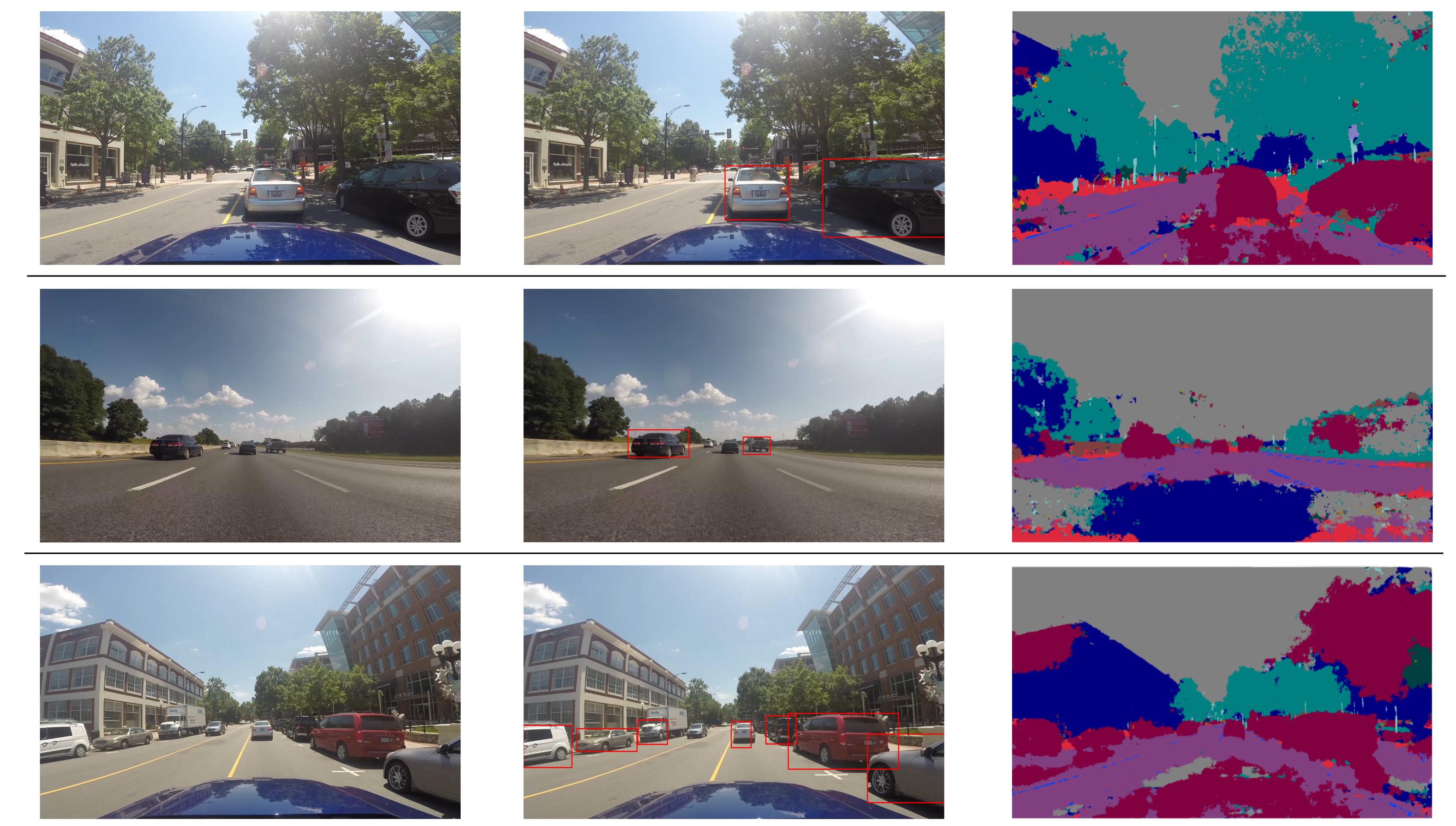
Depth Representation and Fusion



- Multimodal deep learning can serve as a data fusion component of ADAS systems.
- Reasoning using multiple data modalities has the advantage of teaching the network more robust representations
- Using a NN for data fusion can replace more computationally expensive solutions such as ensemble networks or traditional fusion solutions

Current Results/Inference Performance

The following results illustrate detection and segmentation performed on a custom data set.



The results shown above are a small subset of the images taken from the custom data set. The left images illustrate the surroundings of the vehicle, the middle image shows bounding box detection, and the right image gives a segmented image.

Accelerator	Execution Time (ms)	Frames Per Second (fps)
Jetson TX1	~710 / N/A	~1.5 / N/A
K20	~110 / ~310	~9 / ~3
K40	~90 / ~280	~11 / ~3.6
Titan X	~75 / ~110	~13 / ~9

Execution time and FPS based on ZF/VGG16 networks for bounding box detection (Faster RCNN)

Conclusions & Future Work

Following the methods illustrated by other work, we plan to utilize a combination of the ideas along with depth fusion to build a suite of software for perception in autonomous driving including detection, segmentation, and the depth estimation.
In the long term, there is also a need to move more towards a real-time system. For this reason, many techniques (not limited to network compression) can be employed to create faster inference times for detection, segmentation, or any other deep learning inference.

Inference	Execution Time (ms)	Frames Per Second (fps)
Faster R-CNN	~710	~1.5
DetectNet	~152	~6.6
MM_DetectNet	***	***

Execution Time and FPS based on DetectNet network for bounding box detection (with TensorRT)

The authors would like to acknowledge the support of Dr. Melissa Smith (Clemson University) as a faculty advisor as well as Dr. Andre Luckow (BMW) for partially funding this research