Failure-Resilient Distributed Deep Learning Inference from Edge to Cloud

Siddartha Devic, **Brian Nguyen, Alan Liao, Ashkan Yousefpour, Prof. Jason Jue**
Department of Computer Science, The University of Texas at Dallas

Distributed Neural Networks

As neural networks (NNs) are being employed on more complex tasks, the required number of layers to retain reasonable accuracy has increased. Since the computational complexity of NNs depends on the number of layers (and number of perceptrons per layer), it has become undesirable to store entire NNs in the cloud for inference. Therefore, [2] propose distributing individual layers over IoT, Edge, Fog, and Cloud nodes to speed up inference and increase efficiency of data transmission.

Multiview Camera Dataset

We utilize the multi-view multi-class detection dataset as an IoT benchmark for failure resiliency. We show that even with multiple IoT, edge, or fog node failures, such as a camera or server, we can still perform well (e.g. retain accuracy) in a non-trivial classification task.

The dataset contains the same street from 6 different viewpoints (**Figure 1**), with object bounding boxes for frames captured from each camera. The dataset itself contains bounding box information for three classes: car, person, and bus. A single data point is then a collection of 6 images of an object from different fixed angles, and an associated label.

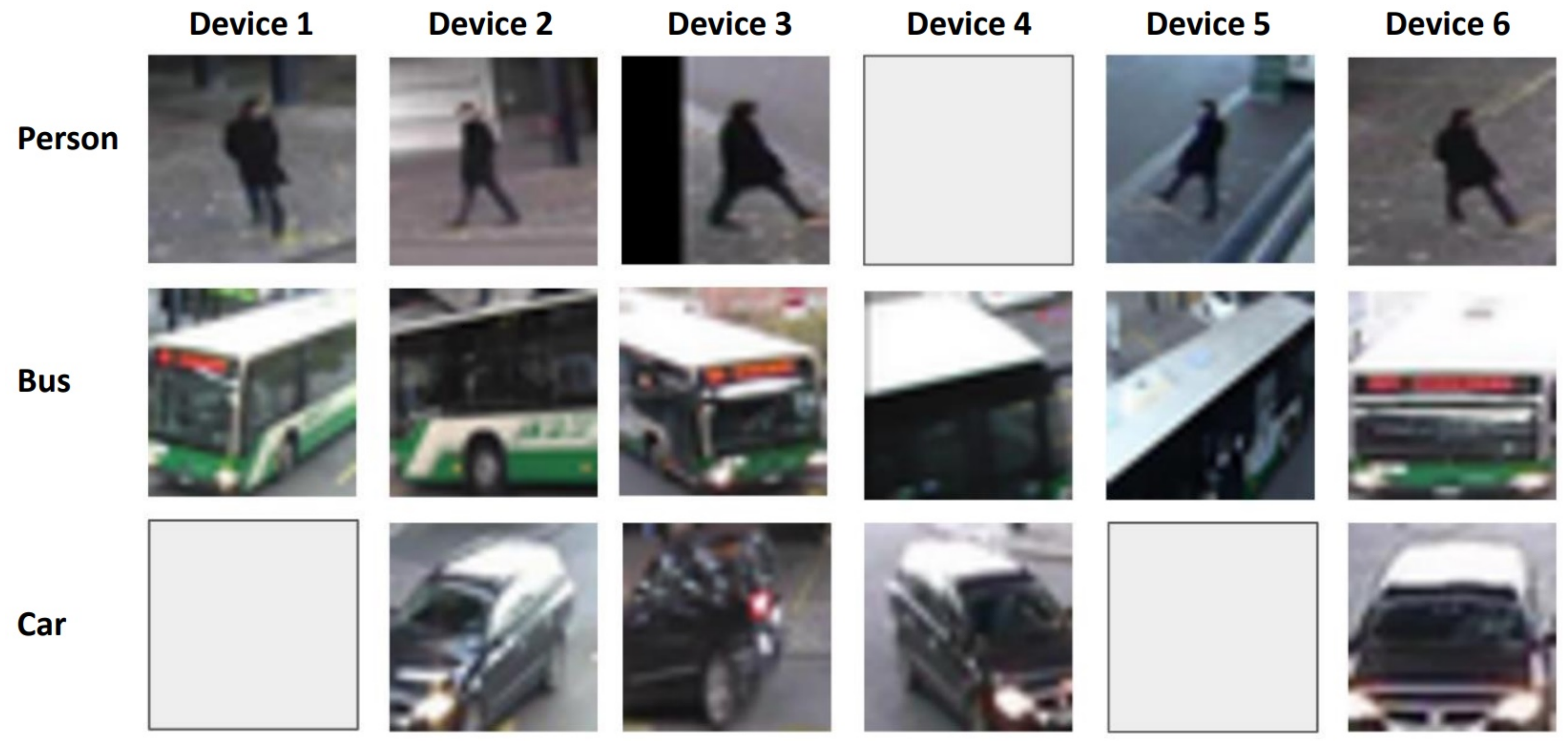

Figure 1: Example data from the Multiview Camera Dataset. Blank images correspond to obscured camera angles.

Measuring Resilience

We construct a weighted failure metric for measuring how resilient a neural network is over a certain physical network architecture. We test three configurations of probabilistic node failure in (**Figure 2**). For example, for the survivability configuration "High", Fog Node 4 fails with probability .99, Fog Node 3 with probability .98, etc. We run our network on a holdout set for each of 2^8 failure configurations, and return the expected accuracy given by weighing the accuracy of a particular configuration with the probability of it happening.

Residual Connections

In a typical neural network, each layer is only connected sequentially. The authors of [1] introduced residual connections, which links a layer in a given level to numerous layers above. It has been observed that these residual networks both increase accuracy and combat gradient inflation, which makes learning larger networks easier.

We propose using residual connections to increase accuracy during partial node failure. Generally, when a physical node in a sequence fails, no information from the entire branch under the node is passed up to the parent if no other connections are used. However, residual connections allow information to be passed around a node both with and without failure.

Our experimental setup (**Figure 2**) investigates how useful residual connections are for a realistic classification task. Each dotted line represents a residual connection between logical nodes (which may contain multiple layers of a neural network). Results are presented in (**Figure 3**).

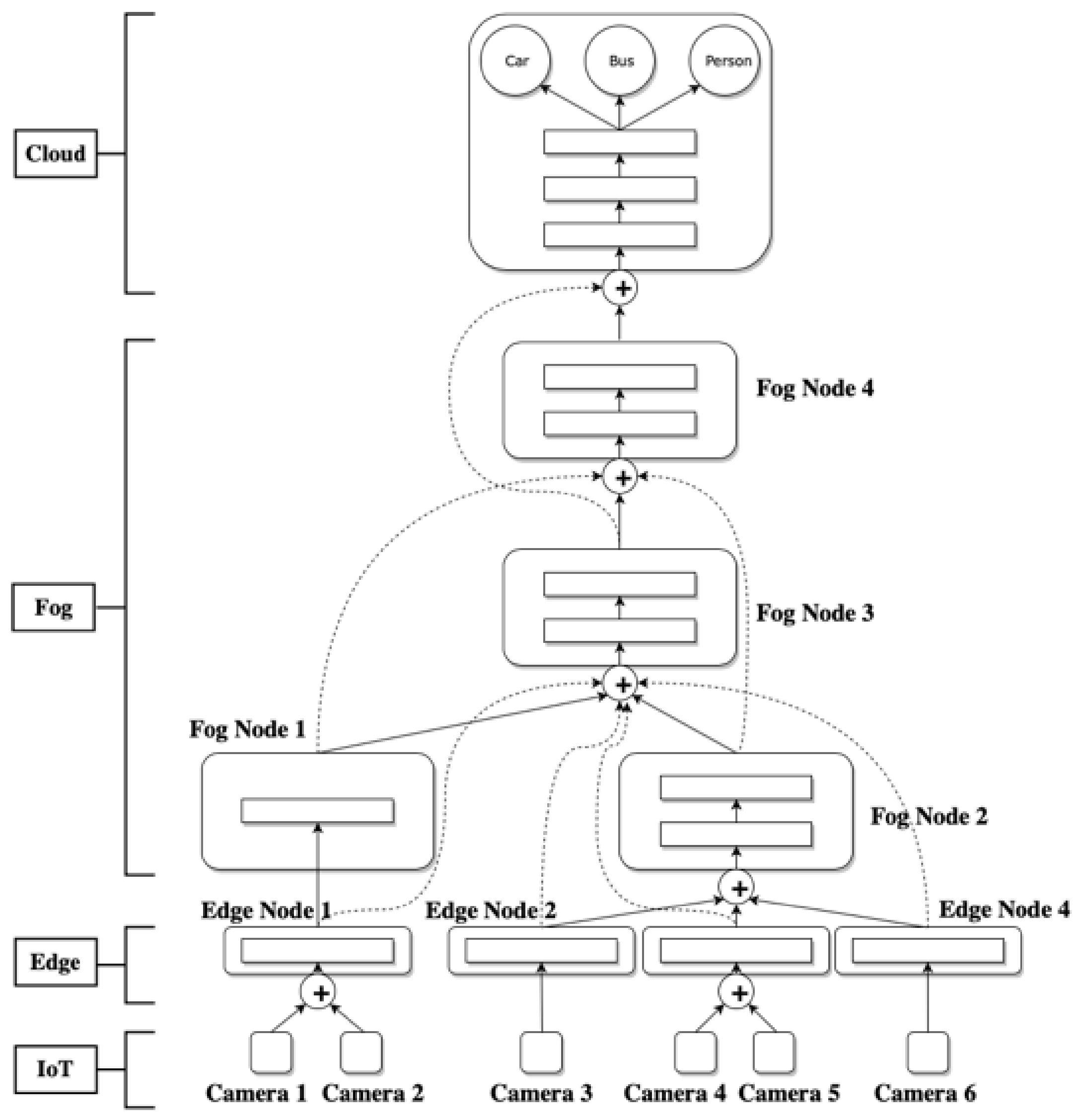

Figure 2: Experimental Architecture

Layer-wise Dropout

The other method we designed for increasing failure resiliency consists of failing a layer during training (i.e. setting its output to 0) with the probability that the layer will fail during inference time. This simulates failure during the training process, and intrinsically makes the neural network more robust to layer-wise failure. We are motivated by Networks of Stochastic Depth which also perform layer-wise dropout, but not for the purpose of failure resiliency.

Results

We present accuracy on a holdout set with three methods, baseline, residual connections, and layer-wise dropout. Each column corresponds to a survivability configuration described earlier.

Method	Acc (High)	Acc (Med)	Acc (Low)
Baseline	94.4%	81.3%	71.9%
Residual Connections	96.3%	88.1%	83.5%
Layer-wise Dropout	**96.7%**	**89.8%**	**85.4%**

Figure 3: Results for both methods on the Multi-view Camera Dataset

As expected, both residual connections and layer-wise dropout both work better than a naive distributed neural network in maintaining resiliency in cases of failure. This implies that it may be necessary to change distributed neural network architectures to maintain resiliency in the case of partial or massive failure of physical nodes.

Future Work

Since residual connections represent a physical connection between nodes, We plan to investigate how to find *which* residual connections are the most important to keep accuracy above a threshold. Additionally, we plan to investigate layer-wise dropout during training to determine optimal rates of failure for maximum expected accuracy.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[2] S. Teerapittayanon, B. McDanel, and H. T. Kung. Distributed deep neural networks over the cloud, the edge and end devices. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 328–339, June 2017.