NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

School of Computer Science and Engineering
College of Engineering

Neural Network Compression Techniques for Out-Of-Distribution Detection

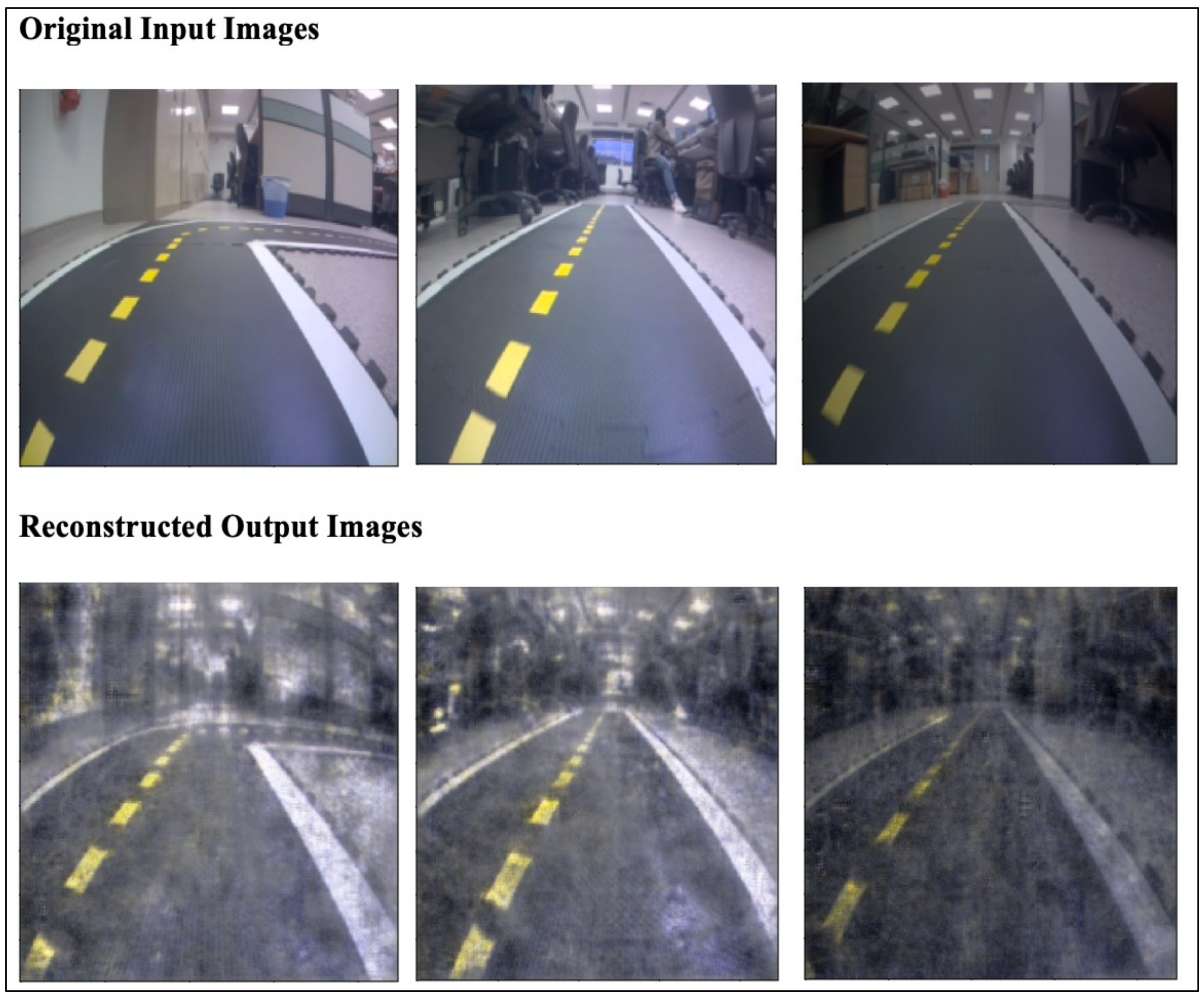
SCSE 21-0449

Student: Bansal Aditya **Supervisor**: Assoc. Prof. Arvind Easwaran

Project Overview

One of the key challenges in deploying ML models on embedded systems are the numerous resource constraints, for instance, memory footprint, response time, and power consumption. Such real-time systems require resource-efficient models with low inference time while maintaining reasonable accuracy. In the context of OOD detection, despite the detection model having a high classification accuracy, if the inference time is too high, the system might be rendered ineffectual.

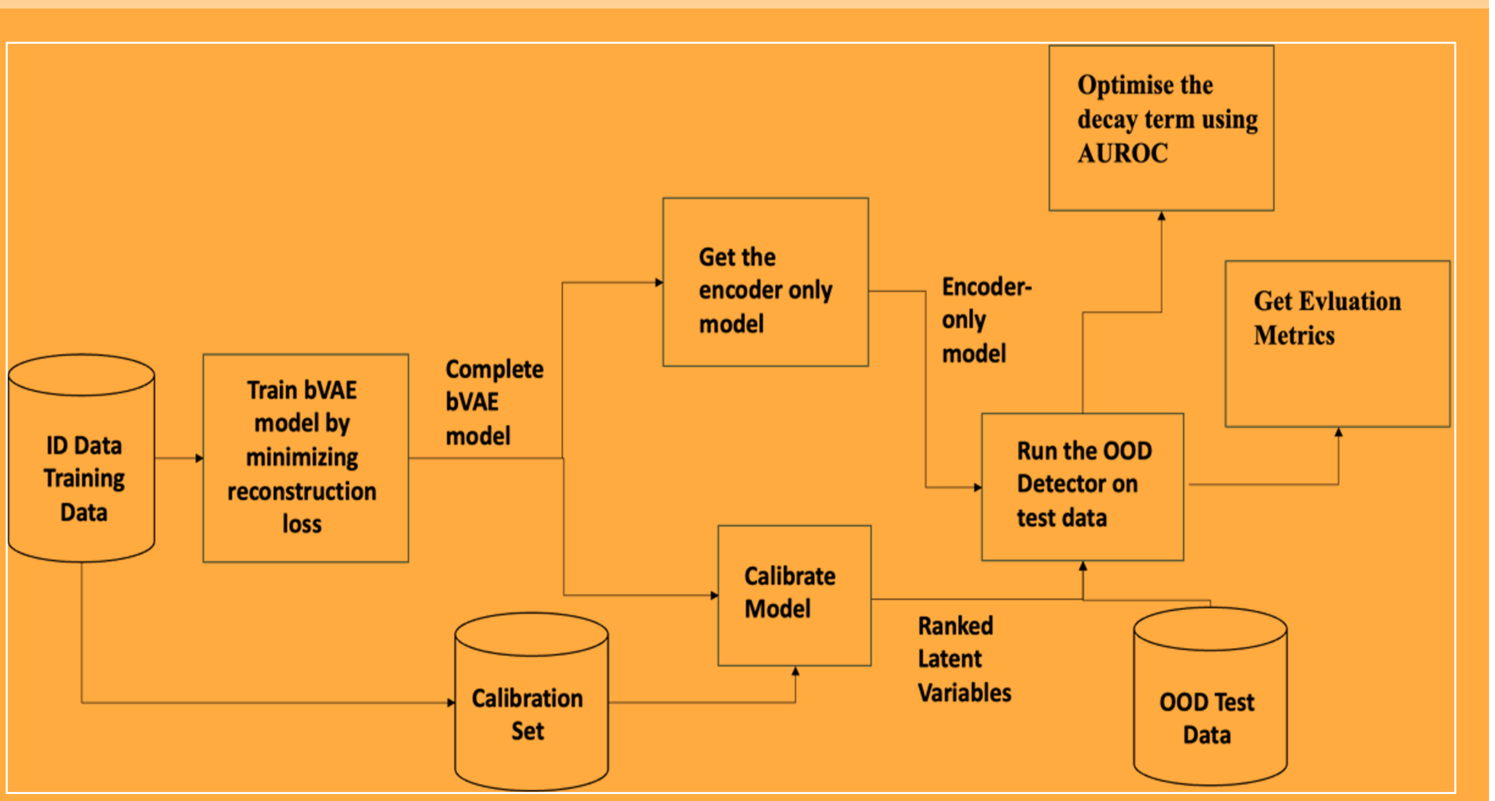
By implementing the above techniques on a real-time embedded system of DuckieBot, we studied the performance of these methods, particularly for the task of OOD detection. The compression techniques of pruning, quantization, and knowledge distillation have been experimented with, and analyzed on numerous metrics, for execution time, memory usage, reconstruction loss, and OOD metrics like ROC curve, True Positive, and False Positive Rates.


Original Input Images

Reconstructed Output Images

OOD Detection Pipeline

1. **Training the β-VAE Model : The VAE model is first trained on the in-distribution data gathered in different lighting conditions**
2. **Get encoder-only model : The weights of the decoder part are discarded, and the encoder-only model is used for the OOD detection**
3. **Calibrate Model : The encoded latent variables are ranked based on their amount of variance for the corresponding partition**
4. **Run the OOD Detector : The OOD detector is then run on the test set using the encoder-only bVAE model and the calibration file**
5. **Optimize the decay term : The resulting files from the OOD detection are then used to calibrate the decay term of the OOD detector**
6. **Get Evaluation Metrics : Finally, the evaluation metrics for accuracy, execution time, memory footprint, and OOD performance are recorded**


ID Data Training Data

Train bVAE model by minimizing reconstruction loss

Complete bVAE model

Get the encoder only model

Encoder-only model

Optimise the decay term using AUROC

Get Evluation Metrics

Run the OOD Detector on test data

Calibrate Model

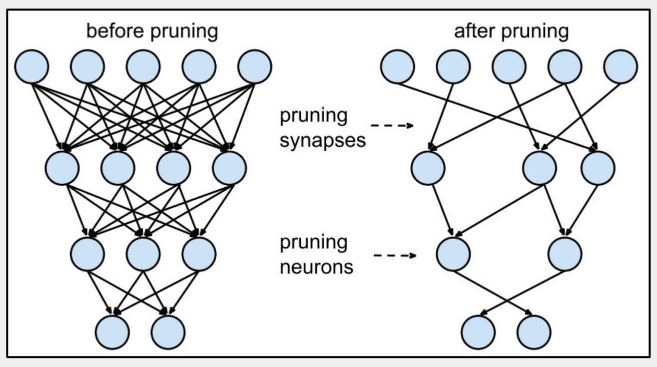
Ranked Latent Variables

Calibration Set

OOD Test Data

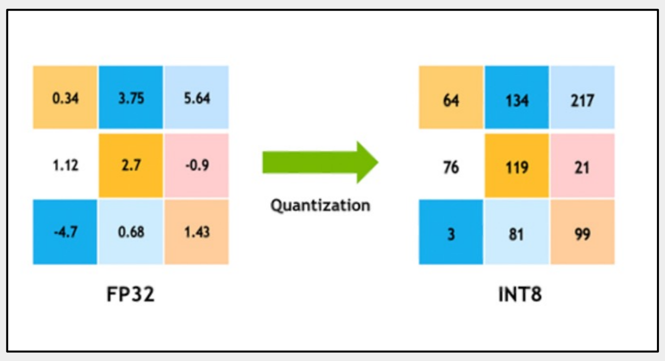
Pruning

- Neural network compression by pruning the network weights is one the most popular techniques, which is commonly used to reduce the number of parameters in a pre-trained network
- The fundamental idea is to remove the redundant and inconsequential parameters from a neural network



Quantization

- The compression technique of quantization has proved to be successful in both training as well neural network inference, in research and industrial settings
- Broadly quantization can be categorized into three different algorithms, namely dynamic quantization, post-training static quantization, and quantization aware training.


FP32

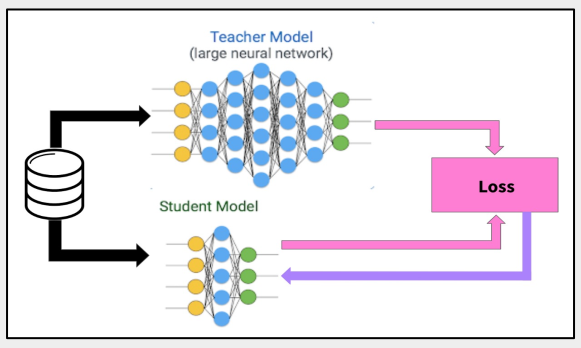
0.34	3.75	5.64
1.12	2.7	-0.9
-4.7	0.68	1.43

INT8

64	134	217
76	119	21
3	81	99

Knowledge Distillation

- Knowledge Distillation aims to compress neural networks by 'distilling' the knowledge from a larger complex model or an ensemble of models to a simpler and smaller model, with minimal loss in accuracy
- Knowledge distillation is typically performed on complex neural network architectures which might potentially be overparameterized.


Teacher Model
(large neural network)
Student Model
Loss