Evaluation of object classifier and cross-layer network emulator: Application to autonomous vehicle

by

Hamad Ullah

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:
Tichakorn Wongpiromsarn, Major Professor
Haridesh Rajan
Jia Liu
Kris M De Brabanter

Iowa State University

Ames, Iowa

2022

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, Dr. Tichakorn Wongpiromsarn(nok) for her guidance, patience, and support throughout this research and the writing of this thesis. Her insights and words of encouragement have often inspired me and renewed my hopes for completing my graduate education. Moreover, her dedication to the students' success is phenomenal. She encouraged me to write a paper and seek my utter potential. Her insights on the paper helped my writing to be much more effective. I would also like to thank my committee members for their efforts and contributions to this work: Dr. Jia(Kevin) Liu, Dr. Kris Debrabenter, and Dr Haridesh Rajan. I would also like to extend my gratitude toward some of the amazing people I got to work with during my master. Without their support this project could not have been possible. First of all, I would like to say my gratitude to Weisi Fan, who I knew as a TA and become someone I respect and care. His continuous support anyone would ask for in a teammate. I would also like to say my gratitude to Gokul, Renai, Nihaal, and Shuhan.

Last but not least I would like to thank my parents, family, and God for keeping me sane. Without their support and guidance I wouldn't be here.

ABSTRACT

Semantic understanding of the perceived world is the key element in autonomous systems' decision-making. The perceived world relies on object classification of each class probability in a given scenario. The widely used approach is the maximum of all class probabilities output of the probability distribution is associated with the object or image in a classification task. A more sophisticated system may also utilize the probability distribution over the classes instead of basing its decision only on the most likely class. The current best approach relies intensively on the accuracy of the model. However, when presented with new data, the model fails to obtain 100% accurate results. Inaccurate classification leads to unsafe behavior in an autonomous system. Let us take an example of an autonomous system. Suppose there is a pedestrian on the crosswalk. The system will only stop if it correctly classifies the object on the crosswalk. Autonomous vehicles need multiple of sensors and actuators. Running wires to connect all sensors and actuators increases the level of complexity.

This thesis presents a new task-based neural network for object classification and compares its performance with a typical probabilistic classification model to improve threshold-based probabilistic decision-making. Furthermore, it introduces a configurable cross layer network emulator known as EMANE with wireless communication optimization for the distributed system based on HeavyBall optimization.

CHAPTER 1. GENERAL INTRODUCTION

1.1 Motivation

Car manufacturers have widely used adaptive cruise control. Even entry-level newer models come with some drivers' safety features. These safety features include lane keep assist, collision mitigation, and a warning system to keep drivers attentive and increase safety. Such partial autonomy paves the way for research in fully autonomous vehicles, which do not require human intervention. Academics and industries have been extensively working on fully autonomous vehicles. There are six levels of vehicle autonomy, level 0 being the lowest and 5 being the highest by SAE standards. Fully autonomous vehicles are known to be in level 5 of SAE standards. Perception and control are the two main components of the autonomous system, as shown in the figure 1.1. Such systems consist of sensors relaying information about the surrounding

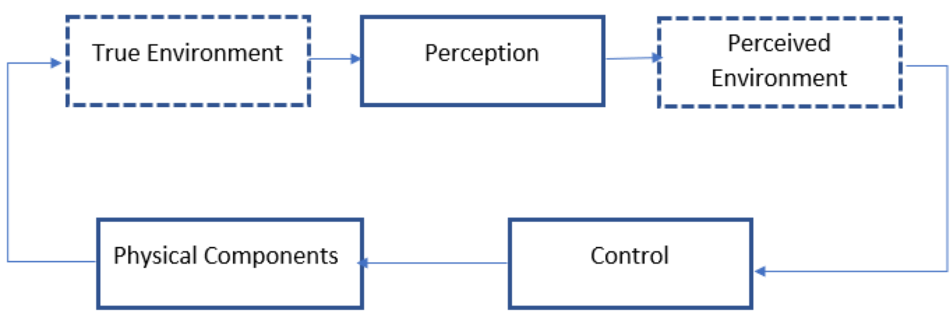


Figure 1.1 A typical architecture of autonomous systems.

environment to the controller. The control system with computational capabilities then executes the desired motion of the vehicle based on the perceived surrounding environment provided by the sensors. The desired motion is often defined within mission goals such as traffic rules, collision mitigation, maintaining lane, etc. Early autonomous systems were heavily reliant on expensive sensors such as LIDAR to provide accurate environmental perception [3]. Figure 1.2 shows the working of the LIDAR technology. Alternative to LIDAR technology, cameras mounted on the top of the windshield and deep learning models, especially convolutional neural networks (CNN), have been widely used in the perception component of many autonomous systems. These models take samples drawn from the whole population to perform the statistical inference in the training process. For example, a model for handwritten digit recognition can be constructed from the MNIST dataset. MNIST dataset contains samples drawn from the handwritten digits data.

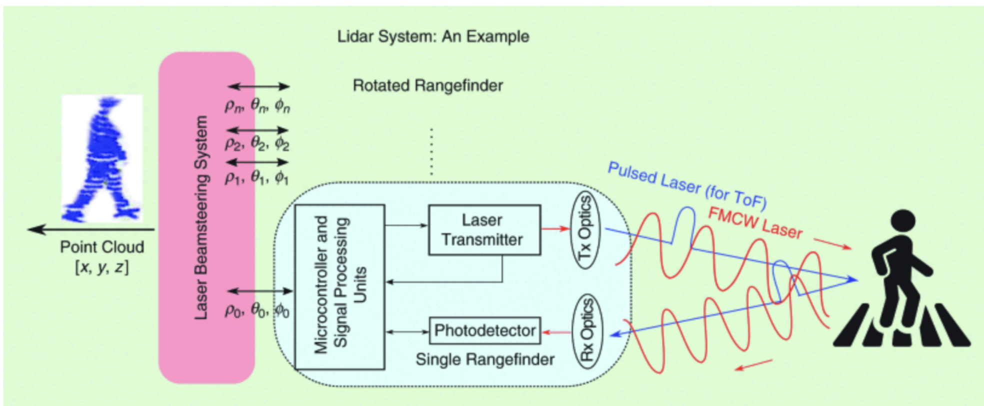


Figure 1.2 An example of a ToF laser rangefinder [4]

A model is, therefore, only an estimator obtained from the samples. As a result, it is impracticable for a deep learning model to make a correct prediction on all new data. In particular, most existing models cannot achieve 100% accuracy in real-world classification tasks. Motivated by failure with the perception component in the autonomous system, in this thesis, we proposed a task-based network, a new approach to object classification task, and a research tool based on heavyball optimization for the researchers to conduct network analysis of various networking scenarios in the autonomous vehicles using our tool in Extendable Mobile Ad-hoc Network Emulator (EMANE).

1.2 Related Works

Deep learning models have been widely deployed in the perception component of the autonomous vehicle for semantic segmentation and object detection [2]. The softmax function is the popular choice for the classification task in deep learning classification tasks strictly as the last layer. However, the softmax function is the exponential function, and small additions to the softmax inputs can lead to a substantial change in output [1]. By treating the output of the softmax as the probability of the object being associated with a particular class, the controller may treat the maximum probability belonging to that class in the decision-making process. Another widely used technique in decision making is threshold t_c is introduced to associate the probability of an object being in a particular class if the probability is at least threshold t_c. In a threshold-based decision-making approach, the object may belong to multiclass. As a result, the output could be ambiguous, and the controller might take the conservative approach. In this thesis, we present a new task-based network structure to be utilized by such a controller. Instead of explicitly computing the probability, our network directly determines whether the probability is above or below a given threshold

1.3 Thesis Overview and Contributions

The objective of this thesis is to analyze the perception component and introduce EMANE for applications in the autonomous system. The research presented here has two key components - Design of the perception component and wireless communication research tool for radio frequency and optimization in communication between different components. Perception component is often comprised of some visual representation of the surrounding with some deep learning model for semantic analysis. Visual components often used are lidar and camera module. Lidar uses laser technology and generate point cloud of different object around the car and camera module take constant recording of the surrounding for further analysis. This contribution was due to the observation that control component expects the perception component to be 100% accurate. If perception component fails to observe most important class during semantic analysis, the control component will behave according to the perceived environment. The analysis of perception component is through the behaviour of deep learning with given input.

1.4 References

[1] Hendrycks, D. and Gimpel, K. (2016). A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

[2] Jebamikyous, H.-H. and Kashef, R. (2022). Autonomous vehicles perception (avp) using deep learning: Modeling, assessment, and challenges. *IEEE Access*, 10:10523–10535.

[3] Kuutti, S., Bowden, R., Jin, Y., Barber, P., and Fallah, S. (2020). A survey of deep learning applications to autonomous vehicle control. *IEEE Transactions on Intelligent Transportation Systems*, 22(2):712–733.

[4] Li, Y. and Ibanez-Guzman, J. (2020). Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, 37(4):50–61.

CHAPTER 2. INTRODUCTION TO MACHINE LEARNING AND AUTONOMOUS SYSTEMS AS DISTRIBUTED SYSTEMS

Hamad Ullah[1]

[1]Department of Computer Science, Iowa State University, Ames, IA, 50014, USA

Modified from a manuscript to be submitted to *IEEE Journal ICRA*

2.1 Abstract

Object detection relies on the process of humans labeling the images and computers learning a pattern. The computer can learn such patterns using a statistical model or sophisticated algorithm. Such techniques are known as machine learning. This chapter provides background for later chapters and an overview of machine learning and autonomous vehicles as distributed systems.

2.2 Machine Learning Techniques: Object detection

Machine learning is a type of artificial intelligence that enables software to predict new data without being explicitly programmed. The primary objective of machine learning is to gather the observable feature provided in the input data. Based on that input data, the model tries to build a mathematical model to predict the values of the new data. Machine learning can be further divided into categories[5]:

Supervised machine learning relies upon pre-classified, i.e., labeled data, which associates the pattern to the labels learned during the training process. The model then utilizes those learned features to predict, i.e., classifies the image based on the new data features.

Unsupervised machine learning does not utilize the pre-classified data. Instead of learning patterns with specified answers as labels during the training, the unsupervised learning algorithm tries to learn the hidden patterns from the unlabeled data, i.e., finding groups in data (clustering), summarizing the data distribution, etc.

Semi-supervised learning is a middle ground between supervised and unsupervised learning. The subset of the data is labeled, and a subset of the data is unlabeled. This kind of dataset is essential in finding out the uncertainty in the model. Generally, it considers a smaller quantity of labeled data and a larger quantity of unlabelled data.

Reinforcement machine learning interacts with the environment and assigns the error and rewards to the agent. The agent is typically simulating the set of possible actions, and based on a sequence of actions taken by the agent, it assigns the score. This machine learning does not need a dataset; however, a specific set of scoring techniques would correctly identify the appropriate behavior.

A subset of machine learning is known as deep learning, as shown in figure 2.1. Deep learning models have solved the problem that has shown the resistance to the best of approaches[4]. Deep learning model has range of applications such as image recognition [3] [1], speech recognition [2] etc.

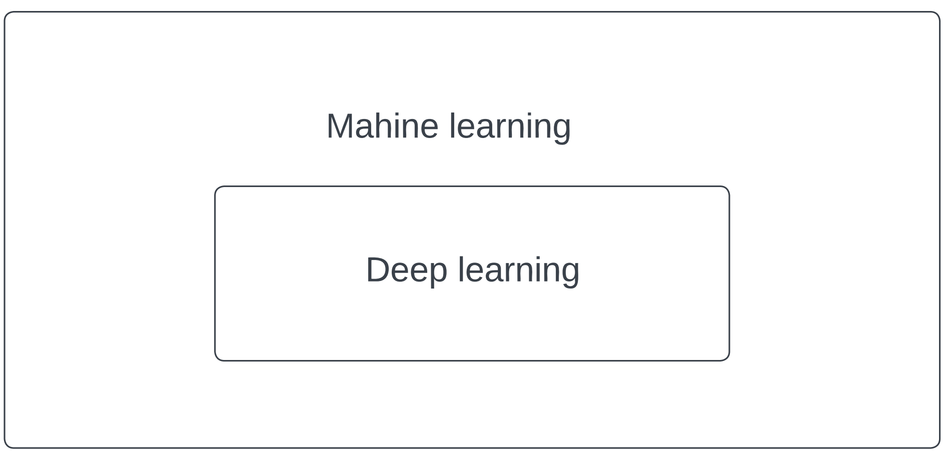


Figure 2.1 Relationship between machine learning and deep learning

2.2.1 Deep Learning

The deep learning model is multi-layer architecture as shown in figure 2.2. The network takes fixed-size input (for example, an image) and outputs the probability p_c as the probability of the image $P_c \in C_i$.To go from one layer to the next, a set of units compute a weighted sum of their inputs from the previous layer and pass the result through a non-linear function [1]. The hidden

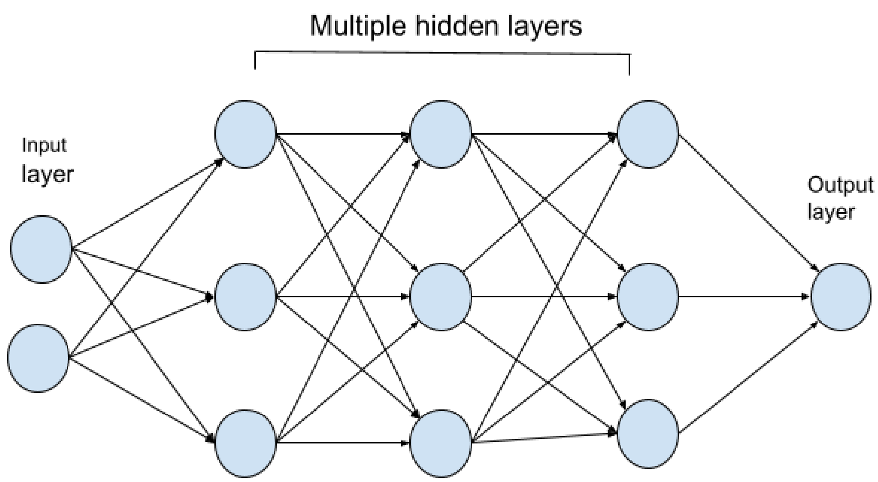


Figure 2.2 Relationship between machine learning and deep learning

layer distorts the input in a non-linear way so that category becomes linearly separable for the last layer. Among many different types of deep learning networks, convolution neural network has gained popularity in pattern recognition.

2.2.1.1 Convolutional Neural network

The CNN network, such as Alexnet in figure 2.3 has the first layer as a convolutional layer. Kernels and strides define the convolutional layer. A kernel is defined as a matrix, i.e., 2x2, 4x4. The kernel uses a sliding window technique to extract the feature from the image. For example, let us consider a 4x3 matrix and kernel 2x2. The figure 2.4 illustrates the convolutional layer computation. Each pixel defined in the kernel 2x2 is summed up together and saved into the new matrix as shown in figure 2.4.

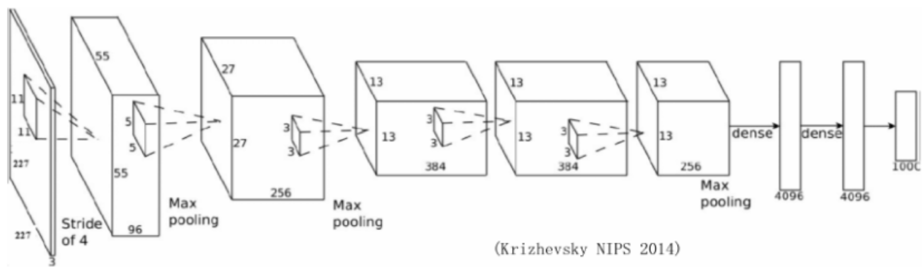


Figure 2.3 Alexnet introduced by krizhevsky 2014

2.2.1.2 Pooling layer

The layer after that is the Pooling layer. The pooling layer works exactly like the convolutional layer; however, the difference is that it either puts the average, maximum, or global of the pixel defined in the kernel.

2.2.1.3 Dense layer

The dense layer is the most commonly used in the neural network model. The dense layer performs matrix-vector multiplication.

This concludes the overview of the machine learning. For our task based network we considered a supervised CNN network which is described in the next chapter

2.3 Communication layers: MAC and Physical layer of wireless network

The OSI model describes seven layers that computer systems use to communicate on the internet. The first layer is the physical layer, followed by the data link, network, transport, session, presentation, and application layers. The session, presentation, and application layers are defined as software layers.

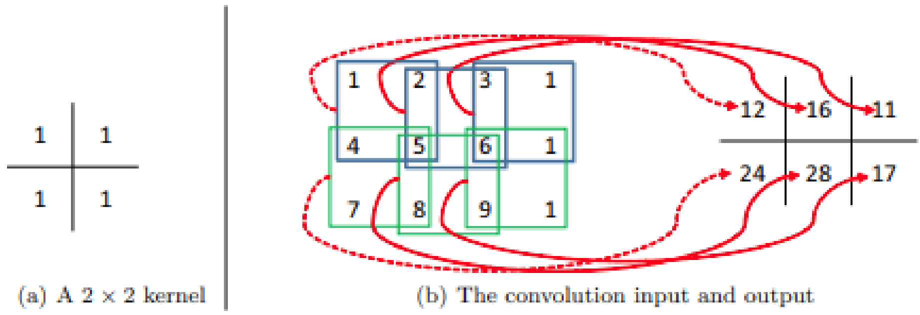


Figure 2.4 Convolutional layer illustration [6]

Replacing the wired communication with wireless communication, the transmitter transmits some range of frequency, i.e., 2.5GHz, 5.0GHz, which is then received by the receiver on the other side of the communication. The three main layers defined in EMANE are Transport, MAC, and Physical layer as defined in 4.2. The physical layer transmits bits over a medium. It decides how much communication is to put in the network. Together with MAC(Media Access Control) implements wireless computer communication. The data is transmitted through some broadband radio waves, such as microwaves.

Chapter 4 will explain EMANE in detail.

2.4 References

[1] Farabet, C., Couprie, C., Najman, L., and LeCun, Y. (2012). Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929.

[2] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97.

[3] Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). Proc. advances in neural information processing systems 25.

[4] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

[5] Saravanan, R. and Sujatha, P. (2018). A state of art techniques on machine learning algorithms: A perspective of supervised learning approaches in data classification. In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 945–949.

[6] Wu, J. (2017). Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 5(23):495.

CHAPTER 3. OBJECT CLASSIFICATION AND TASK-BASED NETWORK: APPLICATION TO AUTONOMOUS SYSTEMS

Hamad Ullah [1], Weisi Fan[2], Tichakorn Wongpiromsarn[2]

[1]Department of Computer Science, Iowa State University, Ames, IA, 50014, USA

Modified from a manuscript published in *IEEE ICRQ 2022*

3.1 Abstract

Object classification is the process of classifying the objects in a given image. Typically, the dense layer using an activation function gives out the probabilities p_i in an vector v belonging to single class. Activation function helps neural network use important information while suppressing irrelevant data points. As the name suggests, the activation function decides when the neuron inside the network should be activated.

3.2 Probabilistic approach

A typical deep learning classification models usually use softmax activation function. The standard softmax function is defined:

$$\sigma(z)_i = \begin{cases} \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}, & \text{if } j > 1 \\ \sigma : \mathbb{R}^k \to (0,1)^k, & \text{otherwise} \end{cases}$$

The softmax function is an exponential function. It normalizes the values if they are greater than 1 so that sum of all the class probability in single classification task is equal to 1. Once the probability distribution of the image or objects o_i in a given image has been determined, the maximum class probability in the vector is returned as the class of the image or the object. For example given an image I the softmax function returns a vector v such that v_i is the probability p

of each class $I \in C$ where C^{d_i} and $d_i \in \{1, ..., d\}$ is a d dimensional multi-class classification task. The maximum probability v_i is the indication of the image being in this particular class C_i. So the C_i class is assigned the I[4]

Softmax function is effective at estimating the probability distribution, however, many have observed that this approach may do a poor job of predicting probability. By treating the output of the softmax function as the probabilities of the object belonging to the associated classes, the controller may treat an object as belonging to the class with the maximum probability from resultant softmax vector.

3.2.1 Threshold based probabilistic approach

Another widely used techniques in descion making is a threshold based approach. Suppose the softmax vector for a give image I, the model checks whether probability p_i is greater than or equal to threshold t_i. It provides another verification step. The problem with this approach can be define in three cases:

Case 1: Consider a vector v an resultant of some classification task. The vector has probability p_i for each class i in the classification task. Consider another vector v_t for threshold such that t_i is the treshold for class i. The problem arises when the threshold probability p_i and p_j is bigger than it's corresponding threshold t_i and t_j where i $\neq$ j. This cause the controller to either have a conflicting decision in the control system if the problem is not serious otherwise the controller might take the conservative approach and completely stop the vehicle.

Case 2: The probability $p_i < t_i$ for i $\in$ 0,...n. Suppose the probability of the pedestrian on the cross walk is lower than the corresponding threshold; In this case the vehicle would not stop and keeps going. For example the vehicle classifies the pedestrian with probability p on the crosswalk. The vehicle will only stop if the $p \geq t_ped$ where t_ped is the threshold for pedestrian.

3.3 Task-based Network

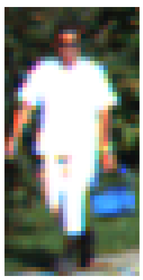
As defined in our paper "Design and Evaluation of Object Classifiers for Probabilistic Decision-Making in Autonomous Systems", Our task-based network is based on the observation that many decisions in autonomous systems are often discrete in nature, e.g., stopping, keeping constant speed, or accelerating for longitudinal control [1]. Thus, it should not require explicit computation of the probabilities. Instead, the actual problem it needs to solve is whether the probability is above a given threshold. As a result, we propose a tasked-based neural network that directly outputs such decisions, instead of having to explicitly compute the probability distribution over the classes. Specifically, a task-based model is defined as a function $M : \mathbb{R}^d \rightarrow \{0, 1\}$, i.e., it is a binary model that only outputs 'Yes' (i.e., 1) or 'No' (i.e., 0) for the task.

3.4 Evaluation of Probabilistic vs task-based

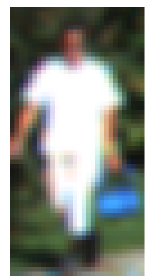
In this section, we use the KITTI dataset [2] to compare the performance of the proposed task-based network and a typical probabilistic classification network that explicitly computes the probability distribution over the object classes. The data used to train the network is the cropped images from the KITTI dataset and contains two classes, pedestrian and non-pedestrian. The label of each image corresponds to the probability that the image belongs to the pedestrian class. For the original images that are cropped from the KITTI dataset, we set the label as 1 if it belongs to the pedestrian class and 0 otherwise. In practice, the data may be labeled by different annotators, who may not always agree on the label, especially when the object is far away or the image is blurry. The probability of each class can then be calculated from the proportion of annotators who label the image with each class. To imitate this process, we use different sizes of Gaussian blur kernel to filter the image. Then, we assign the probability accordingly. Figure 3.1 shows an example of the different level blurred images with the probability. After having the data with its corresponding probability, we can train the network on the data.

Both the task-based network and probabilistic classification networks use the CNN structure and Adam loss function [3]. Before training the task-based network, the label of the images are assigned based on a specific threshold t. For each image x_j, if $p_j > t$ then $y_j = 1$, otherwise $y_j = 0$. After the labels are set up, the task-based network will use 85% of the data to train the network and reserve 15%, randomly selected for the collected data, for evaluation. On the other hand, the probabilistic classification network feeds the training image x_j with its probability p_j into the network to train the model. To avoid the data augmentation, we trained both network with blurred images as well as original images.

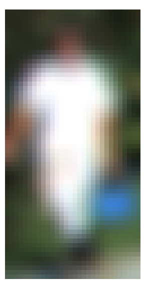
[Original $p = 1$]



[3×3 kernel $p = 0.9$]



[7×7 kernel $p = 0.7$]



[11×11 kernel $p = 0.5$]

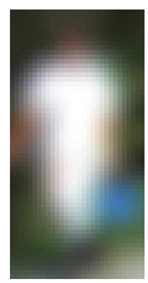


Figure 3.1 Images with different levels of blur

Figure 3.2 and 3.3 show the true negative rate and false positive rate of both networks on the stop decision. From the results of figure 3.2 and 3.3, we can see that both the true-negative and false-positive rates of the task-based network do not vary much with changing thresholds. On the other hand, the results of the probabilistic classification network highly depend on the threshold. For example, with threshold $t = 0.9$, the true negative rate is very high. This observation implies that the predicted probability is not reliable. Moreover, the true negative rates are very high under all thresholds. This implies that even though we feed the network with probability and

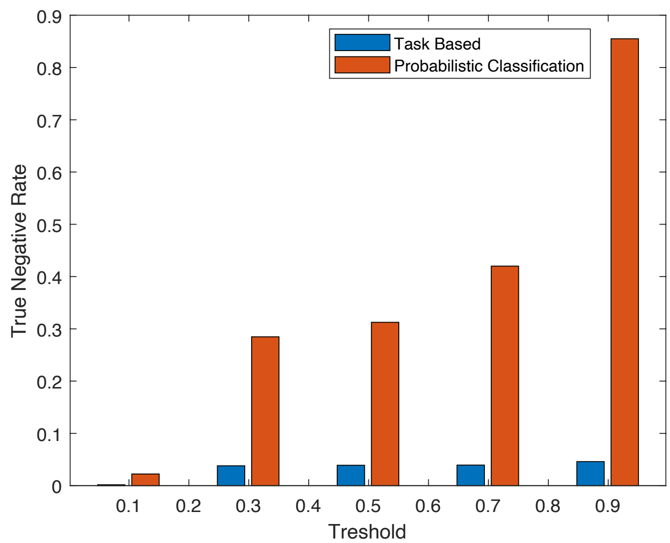


Figure 3.2 False negative rate

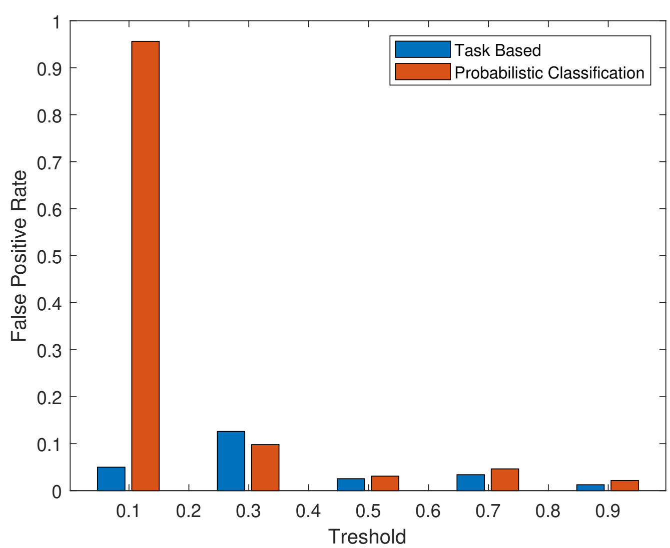


Figure 3.3 False Positive rate

train it, the output probability can still be unreliable. This can be particularly problematic in safety-critical applications such as autonomous vehicles.

3.5 References

[1] Badithela, A., Wongpiromsarn, T., and Murray, R. M. (2021). Leveraging classification metrics for quantitative system-level analysis with temporal logic specifications. In *Proceedings of the 60th IEEE Conference on Decision and Control (CDC)*.

[2] Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*.

[3] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[4] Nwankpa, C., Ijomah, W., Gachagan, A., and Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*.

CHAPTER 4. HEAVY-BALL ALGORITHM IN EMANE: NETWORKING TOOL FOR WIRELESS COMMUNICATION

Jia Liu[1], Atilla Eryilmaz[2], Ness B. Shroff[3], ELizabeth S. Bentley[4]

[1]Department of Computer Engineering, Ohio State University, Columbus, OH 43210, USA

[3]Air Force Research Laboratory, USA

Modified from a manuscript published in *IEEE Journal*

4.1 Abstract

A fast and reliable network is essential for better communication. EMANE is a network emulator that allows the researcher to create algorithms and test them on the platforms. This chapter introduces EMANE and its implementation in EMANE for future research in networking.

4.2 Overview

Defined by US naval research, Extendable Mobile Ad-hoc Network Emulator(EMANE), is a next-generation framework for real-time modeling of mobile network systems. The EMANE components focus on real-time modeling of link and physical layer connectivity so that network protocol and application software can be experimentally subjected to the same conditions that are expected to occur in real-world mobile, wireless network systems. EMANE architecture provides for Network Emulation Modules (NEMs) that can be associated with computer system (real or virtualized) network stacks as interfaces. The EMANE framework further provides an event-driven control bus and logging facilities. Figure 4.2 shows typical architecture of the EMANE emulator.

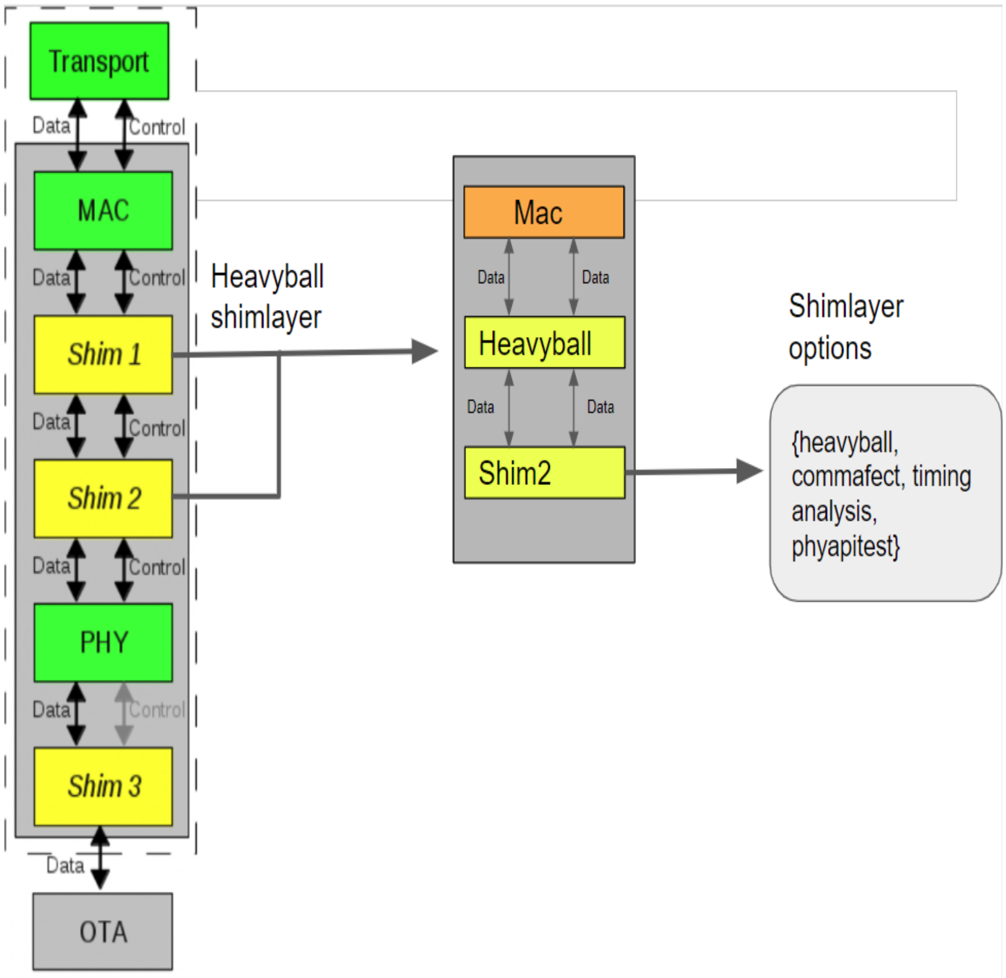


Figure 4.1 EMANE Architecture with heavyball shimlayer

The implementation of Heavy-Ball algorithm[1] in EMANE create an additional features for the researcher to emulate congestion and max weight algorithms [1]. The new architecture does not affect the functionality of MAC and PHY. The new architecture is shown in figure 4.1.

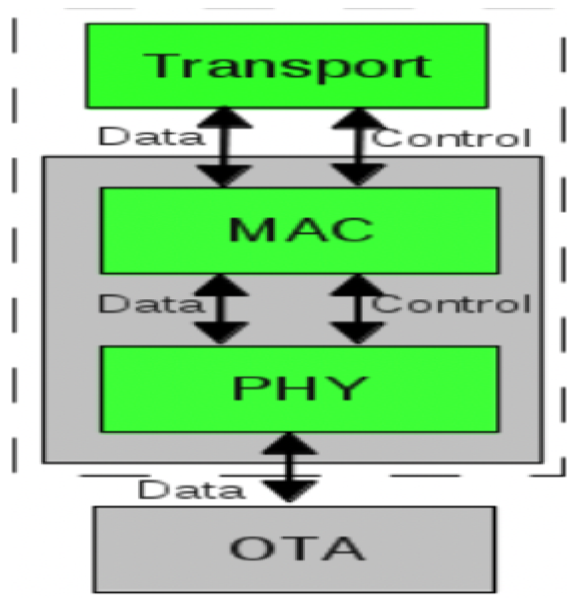


Figure 4.2 EMANE Architecture define by RoboCom

4.3 Application to autonomous vehicles

Autonomous vehicle are designed with many sensors and actuator constantly communicating with each other. With escalating complexity of autonomous vehicle design, internal wiring become extremely intricate and increase the weight putting pressure on steering and brake system [2]. The tool such as EMANE could help researcher emulate some of the communication protocols for autonomous system.

4.4 References

[1] Liu, J., Eryilmaz, A., Shroff, N. B., and Bentley, E. S. (2016). Heavy-ball: A new approach to tame delay and convergence in wireless network optimization. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9.

[2] Wang, J., Liu, J., and Kato, N. (2018). Networking and communications in autonomous driving: A survey. *IEEE Communications Surveys Tutorials*, PP:1–1.

CHAPTER 5. GENERAL CONCLUSION

Although we achieved convincing results, we might want to investigate details of how task-based network performs when tested in real-world scenarios. The next step of this project can be well-versed in testing the task-based network in a real-world environment. For that, we will be using a Duckietown robotics system. Duckietown is an open, inexpensive, and flexible platform for autonomous education and research. We will deploy our evaluation metric and task-based network in Duckietown and test the performance in the real-world scenario. Furthermore, research in v2v networks using different frequency models with EMANE emulators and its implication on data could be done more extensively.