CSE 246 Project Proposal
Explaining the Reality Gap

Oliver Chang
UC Santa Cruz

1 The Problem

Neural Networks (NN) are becoming an increasingly important part of autonomous vehicles. Primarily, they are used to process camera input to identify objects. Cutting-edge NN architectures can achieve impressive results classifying objects from a camera image. However, these models are limited to scenarios they were trained on. For example, a piece of black tape on a stop sign can confuse a NN (Eykholt et al., 2018). One idea to mitigate this issue is collecting more data. This involves driving with sensors collecting information from various road conditions. A challenge with this approach is that it is costly and time consuming. Simulations offer an inexpensive and faster alternative to train NN models.

A simulation environment, like CARLA (Dosovitskiy et al., 2017), can effectively generate image datasets in a wide variety of traffic scenarios, street conditions, and map layouts. Models can then be trained on simulation data and applied in real-life. In robotics, this approach is called crossing the reality gap. The problem with crossing the reality gap is that the models used do not explain their process and final prediction. Models that perform well in simulation might behave differently in a new domain. Figure 1 shows two different simulated mazes where an agent is using a NN to navigate. In this study, the researchers found that the agent performed worse when trained in the low fidelity simulator and applied in the high fidelity maze. This is an example of why NN should be able to explain themselves to help cross the reality gap.

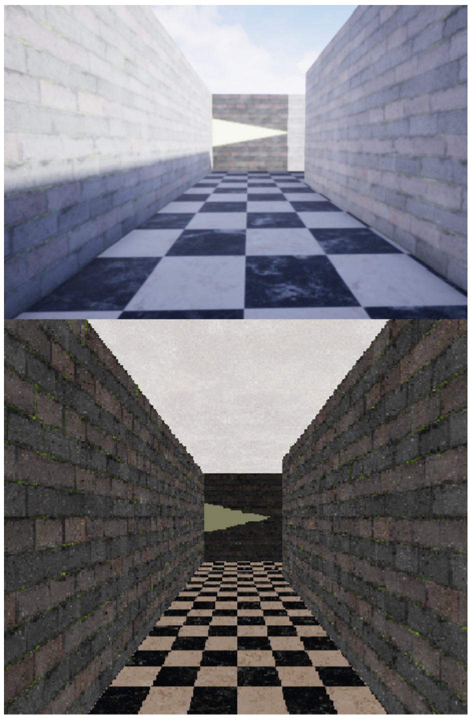
Figure 1: An example of two simulators with different graphic quality but the same maze shot.

2 Problem Matter

While autonomous vehicles have made great strides towards full autonomy, they are still prone to mistakes that could lead to catastrophic results. Tesla, an industry leader in smart car development, had a car on autopilot crash into the rear end of a truck in January 2018 (Valdes-Dapena, 2018). It is imperative that the visual systems trained with NN are safe, robust, and efficient. This will require bountiful and well documented data which is typically challenging to acquire. Simulators can help expedite this process. However, models within simulators are not perfect either and do not adapt in new environments well. Incorporating explainability in the reality gap will improve overall safety in autonomous vehicles.

3 Proposed Solution

Explainable artificial intelligence (AI) is becoming an increasingly relevant field as cutting-edge machine learning models become more opaque. One technique that introduces explainibility in deep NNs is layer-wise relevance propagation (LRP)

(Montavon et al., 2019). LRP has been used for face expression recognition, highlighting cell structures, and finding features in audio source localization. As such, applying LRP in an autonomous vehicle setting has less literature but the problem of classifying objects at a macro-lens remains the same. LRP is a backward-propagation technique used to show which features in an image are relevant. Given a DNN and its prediction, LRP propogates that prediction backwards, calculating a relevance score.

$$R_i = \sum_k \frac{z_{jk}}{\sum_j z_{jk}} R_k \tag{1}$$

Equation 1 computes a relevance score at a specific layer in a NN. The numerator denotes the amount neuron j contributed to neuron k, and the denominator is the sum of all of layers j's contribution to layer k. In other words, the equation calculates a weighted proportion of a neurons contribution divided by the overal layer's contribution. The relevance score, or output, is then used to generate a heatmap because each pixel's relevance is measured. A heatmap can help highlight which pixels were most prevalent in the model's classification prediction.

In this project, I propose developing two end-to-end NN models: one trained on synthetic data and the other trained on real-life data. The simulation can be generated with CARLA, and it would need to resemble a real-life driving scenario as closely as possible for comparison. The second model is trained on real-life 2D images, classifying objects. I plan on training the models so that they classify pedestrians and cyclist from 2D RGB images. nuScenes is a public autonomous driving dataset that provides RGB camera images (Caesar et al., 2019). CARLA is also multi-modal and offers 2D camera images. After, I will apply LRP to the two models and examine which features were important in the simulator and real-life. It is important to note that LRP and its heatmapping methods give global explanations (Samek et al., 2017). So even though the models are concerned with classifying a particular object, LRP will tell us which pixels in an entire scene were relevant.

Explainability through the heatmmaps produced from LRP can be assessed through a framework proposed by (Samek et al., 2017). The idea behind this assessment is that perturbations applied to relevant pixels should worsen the value of $f(x)$, the prediction. Heatmaps tell us an array of pixel-wise relevance scores (R_i). So, this framework picks a relevant pixel and replaces all pixels in an $m \times m$ neighborhood from a uniform distribution. These perturbations and their impact on $f(x)$ can be quantified through a value called an *area over most relevant first perturbation curve* (AOPC). Simply put, the larger the AOPC, more impactful the perturbations were towards affecting $f(x)$. In this comparison, the models will be assessed through this heatmap evaluation framework. Ideally, the features should be the same within the simulation and real-life. If not, then that suggests the simulator is not representing a feature that is in real-life.

Realistically, I plan on training at least one of the domain's NN and applying LRP to produce a relevance heatmap. While there are an abundant source of annotated datasets, configuring CARLA and a real-life dataset for comparison will be a worthy challenge. I hope to complete this step early in the quarter. Once I settle on a dataset, scenario, and simulator, I will pick a state-of-the-art NN architecture and train it to classify pedestrians or cyclists.

I hope to garner explainability experience through this project. In previous projects, I have dealt with developing NN but never voyaged beyond implementing them. As such, this project will deepen my AI/ML skill set by encouraging me to understand what is actually happening in a NN.

Week three will be spent finding a dataset and setting up CARLA. Weeks four through six are for finding a state-of-the-art NN architecture and training it on data. A pretrained model on a well annotated dataset should yield accurate results relatively quickly. Then, weeks seven through ten will be devoted to applying LRP on the models and evaluating their relevance heatmaps.

References

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2019. nuscenes: A multimodal dataset for autonomous driving.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. 2018. Robust physical-world attacks on deep learning visual classification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1625–1634.

Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. 2019. *Layer-Wise Relevance Propagation: An Overview*, pages 193–209. Springer International Publishing, Cham.

Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. 2017. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673.

P Valdes-Dapena. 2018. Tesla in autopilot mode crashes into fire truck. *CNN Tech*, 24.