**PhyCV-Enabled Lane Line Detection**

**Coding Skills Required: Advanced**

## Goals

The goal of this project is to explore applications of physical algorithms in autonomous driving. Specifically, you will quantify the efficacy of PhyCV as a preprocess for deep neural networks on the task of lane line detection for autonomous vehicles. Specifically, this project aims to compare a benchmark lane line detection deep neural network model with a model that includes a preprocessing step of applying the Phase Stretch Transform (PST) algorithm (part of PhyCV library) for feature extraction. Among other things, the outcome should be a comparison on these two pipelines in terms of accuracy, robustness to perturbations, generalizability, and model size.

## Introduction

Lane line detection is a critical capability for autonomous vehicles. Most pre-trained lane line detection models require large architectures to achieve high accuracy. Phase Stretch Transform (PST), being part of the PhyCV library, is a physics-inspired algorithm for edge detection. Given PST’s computational efficiency and contrast insensitivity, it offers a pathway for improving computer vision on the edge. In this project, we will assess PST's capability to reduce model size while enhancing robustness and generalizability for the task of lane line detection for autonomous vehicles.

## Tasks

**1. Familiarize yourself with the PST algorithm**

Read the PST paper, download the code from GitHub, and understand the algorithm and parameters. Generate the sample results as found in the GitHub repository in a Jupyter Notebook.

**2. Understanding BDD100K dataset**

Explore the BDD100K dataset and repository from Berkeley Deep Drive. Understand data formatting, training, evaluation APIs, and metadata. Create a Colab notebook that uses a PyTorch Dataloader to load images along with their category (daytime, nighttime, fog, etc.). Show images and annotations from each of these categories.

**3. Understanding Lane Line Detection Models**

Explore the BDD100K dataset and repository from Berkeley Deep Drive for training and evaluating lane line detection. Demonstrate inference with a pretrained model, and record accuracy on training, validation, and test datasets. You may use a smaller subset of the data if necessary. This will become your benchmark for further experimentation.

**4. Preprocess BDD100K images with PST**

Apply PST preprocessing to images in the BDD100K dataset. Understand the impact of parameters on edge enhancement. Find a suitable set of parameters for processing across different images. Compare the performance of PST feature extraction across daytime, nighttime, and foggy scenes.

**5. Train a Model on PST Preprocessed Data**

Train the benchmark lane line detection model on the PST preprocessed BDD100K images. Are you able to get back to the same accuracy as the benchmark?

**6. Train a Model on PST Preprocessed Data**

Train a lane line detection model with less parameters than the benchmark on the PST preprocessed BDD100K images. Are you able to get back to the same accuracy as the benchmark?

**7. Compare PST+DNN on Night Time Data**

Compare the performance of the PST preprocessed model with the benchmark on specific environment conditions, including nighttime and fog. How do the different techniques compare in terms of robustness?

## Deliverables

PowerPoint slides, final presentation, Colab notebook, and GitHub repository.

## Resources

## Contact

Yiming Zhou (Ph.D. student at Jalali-Lab): yimingz0416@g.ucla.edu

Callen MacPhee (Ph.D. student at Jalali-Lab): cmacphee@g.ucla.edu