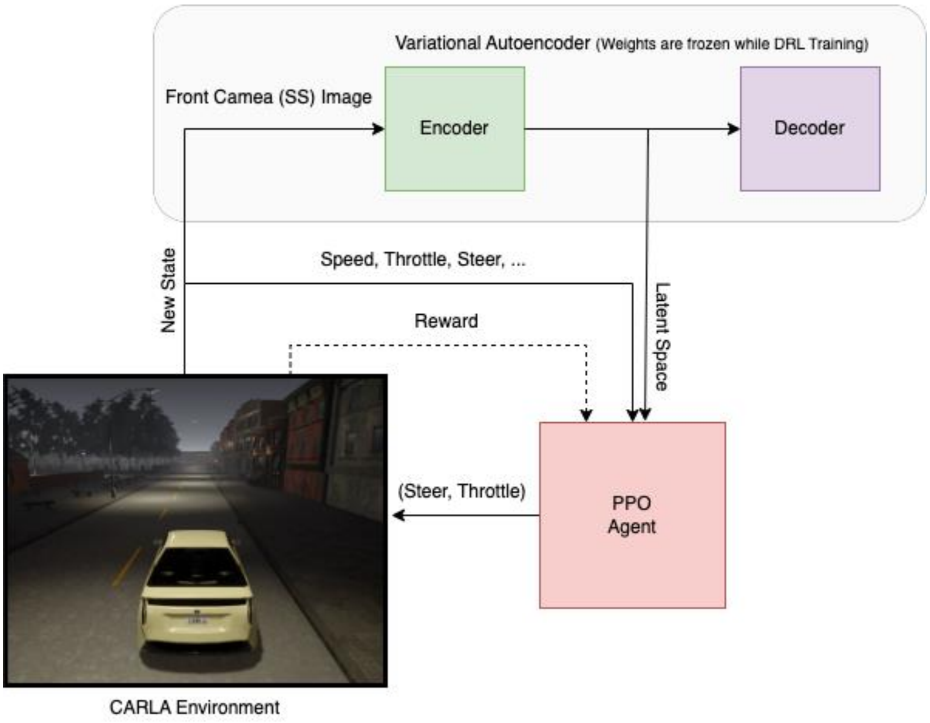
INTRODUCTION

This work focuses on leveraging cutting-edge artificial intelligence techniques, particularly Deep Reinforcement Learning (DRL), to train agents for autonomous driving. The research utilizes the CARLA open-source simulator, which offers a highly realistic urban setting for model training, circumventing the risks and ethical concerns associated with testing raw algorithms directly in real-world scenarios.

ARCHITECTURE

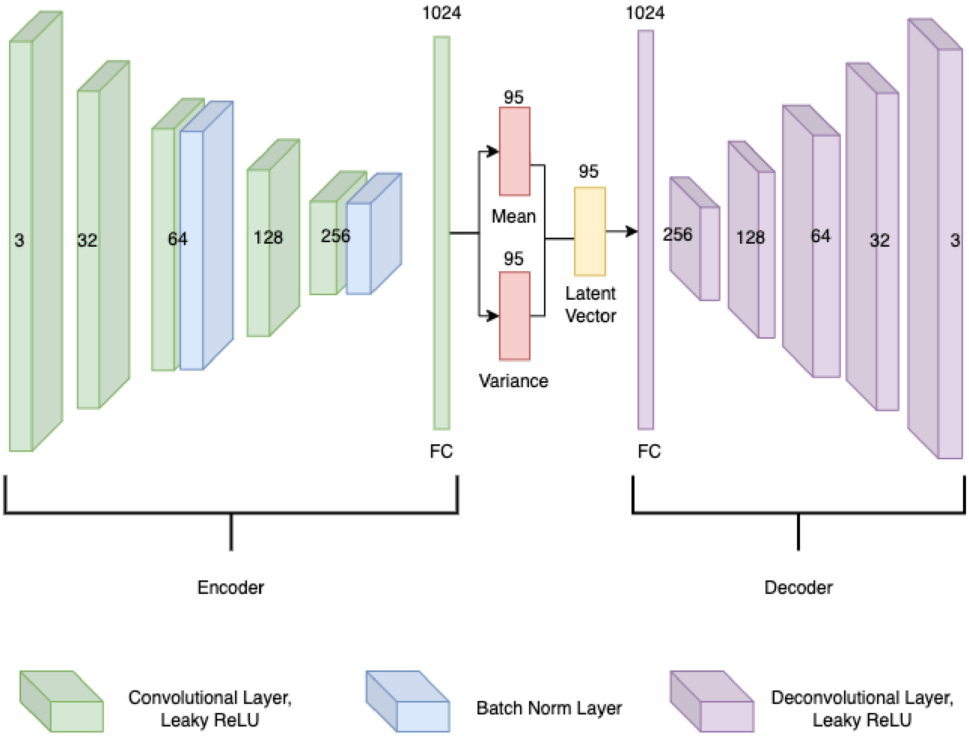


Architectural Methodology

Algorithm Used

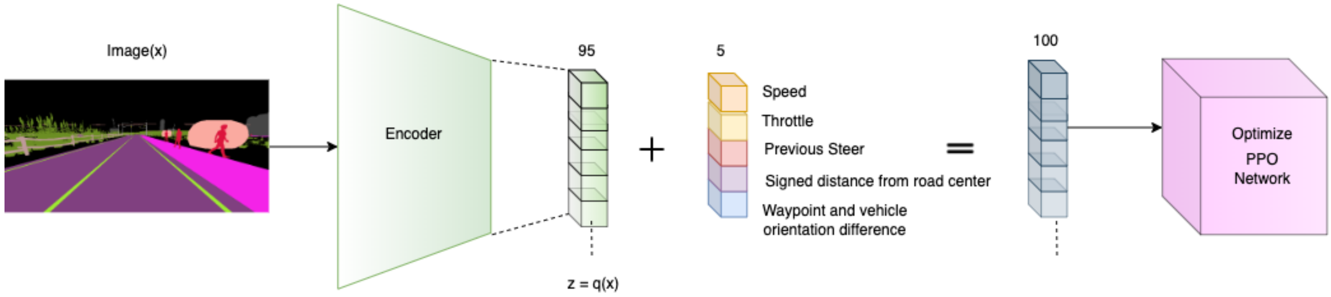
PPO: - We chose Proximal Policy Optimization (PPO) as the deep reinforcement learning algorithm for continuous control that worked best in our tests. PPO is a model-free reinforcement learning algorithm based on policy gradients that stops divergence with a first-order trust region criterion. In this part, we will outline the specifics of our PPO implementation.

Variational Autoencoder: - The Variational Autoencoder (VAE) training process starts by driving around automatically and manually, collecting 12,000 160x80 semantically segmented images we will be using for training. Then, we will use the SS image as the input to the variational autoencoder (h $* w * c$ = 38400 input units). VAE's weights are frozen while our DRL network trains.



Variational Autoencoder

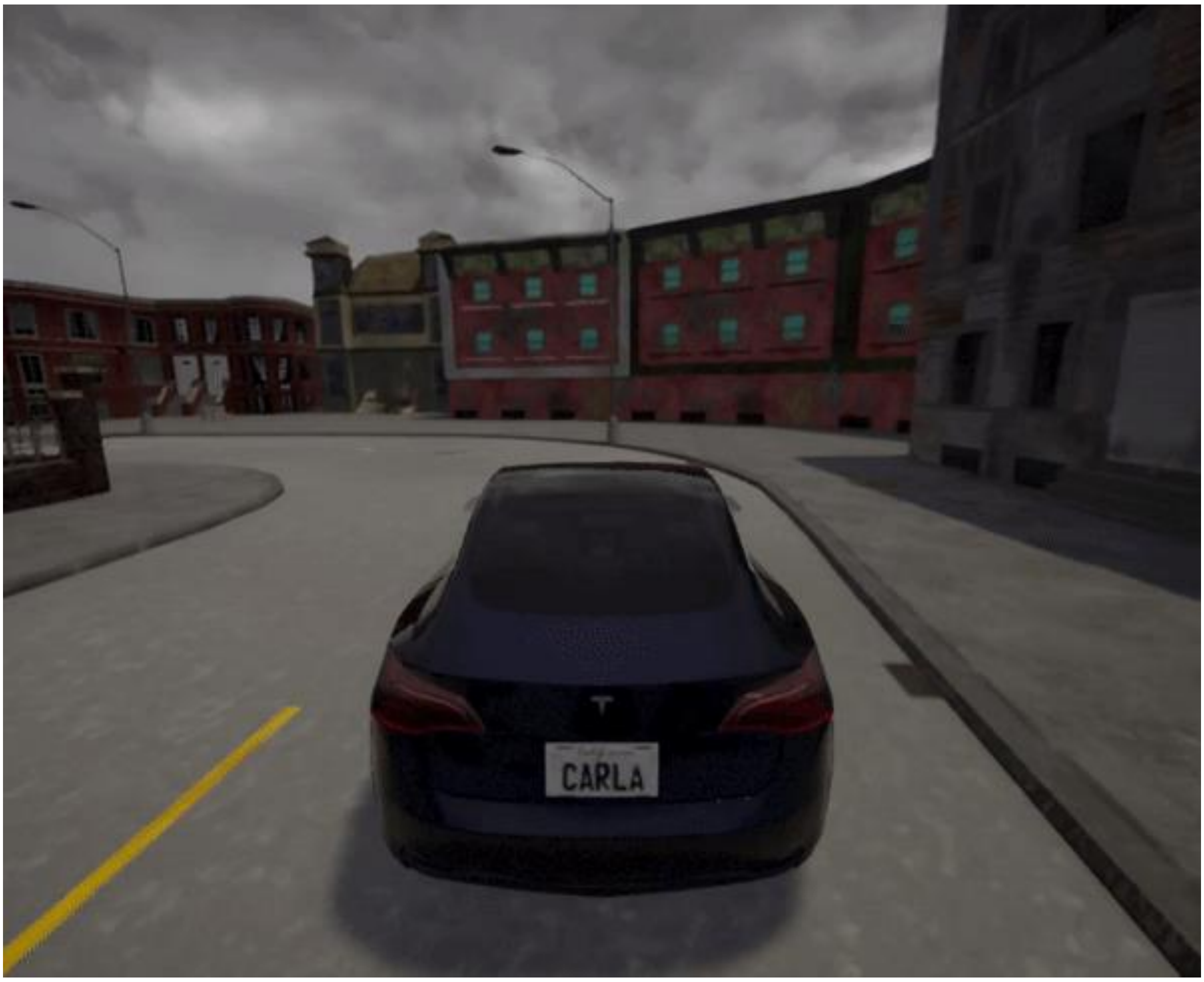
Project Architecture Pipeline (Encoder to PPO)



VAE + PPO training pipeline

Requirements: - Python 3.7, Carla (0.9.8), Nvidia Quadro P5000 with 6 GB of video memory, a 4-core CPU, and 12 GB of RAM,

How Training looks like



Twon 2

Twon 7