A Project on Imitation Learning

Parth Shinde
dept of Computer Engineering
University of California, Riverside
pshin022@ucr.edu
SID: 862466119

Giridhar Peddi
dept of Electrical and Computer Engineering
University of California, Riverside
gpedd002@ucr.edu
SID: 862395129

***Abstract*—This project explores the development of a robust end-to-end autonomous driving system utilizing advanced neural network models, specifically Xception and Vision Transformers (ViTs). Leveraging the CARLA simulator for data collection and augmentation, we aimed to create a scalable and efficient training pipeline. Despite initial challenges, our evaluations revealed the potential of these models in handling diverse and complex driving scenarios. However, due to limitations in resources and time, we were unable to achieve the desired performance. Future work will focus on refining the models, expanding the dataset, and conducting comprehensive evaluations to enhance the robustness and reliability of the autonomous driving system. With adequate resources, we believe these advancements could significantly contribute to safer and more efficient autonomous vehicles.**

***Index Terms*—Github repo link to our project – https://github.com/parthshinde1221/EE260-carla**

I. Introduction

The project is driven by the evolving landscape of autonomous driving technology, which has seen a significant shift from modular approaches to end-to-end methods. This shift is largely due to advancements in robust and powerful models, leading to a resurgence in the application of imitation learning for autonomous driving. Traditional modular approaches, which break down the driving task into separate components, are being outpaced by end-to-end methods that streamline the process by directly mapping sensor inputs to driving actions. Imitation learning, in particular, has gained renewed interest. This method involves training models by mimicking the behavior of expert drivers, leveraging advancements in neural networks to enhance the performance and reliability of autonomous vehicles. The project's motivation is rooted in harnessing these advancements to develop a custom architecture for end-to-end autonomous driving, with the goal of achieving more intuitive and efficient vehicle control. The project introduces a custom architecture designed specifically for end-to-end autonomous driving. This architecture is built upon the principles of conditional imitation learning, which allows for more nuanced and adaptable driving strategies by incorporating conditional inputs (such as different driving scenarios and commands). The primary contribution is the integration of a multi-modal input system into the architecture, enhancing the vehicle's ability to interpret and respond to a variety of driving conditions and commands.The envisioned architecture aims to revolutionize autonomous driving by employing intuitive design principles based on conditional imitation learning. This approach not only simplifies the learning process but also improves the vehicle's performance in diverse driving scenarios. By introducing multi-modal inputs, the architecture can process and integrate various types of sensor data, leading to more accurate and reliable driving decisions.

The project aspires to push the boundaries of what is currently achievable in autonomous driving technology, paving the way for safer and more efficient autonomous vehicles. Through rigorous testing and iterative improvements, the team aims to demonstrate the feasibility and advantages of their proposed architecture, contributing valuable insights and innovations to the field of autonomous driving.

Identify applicable funding agency here. If none, delete this.

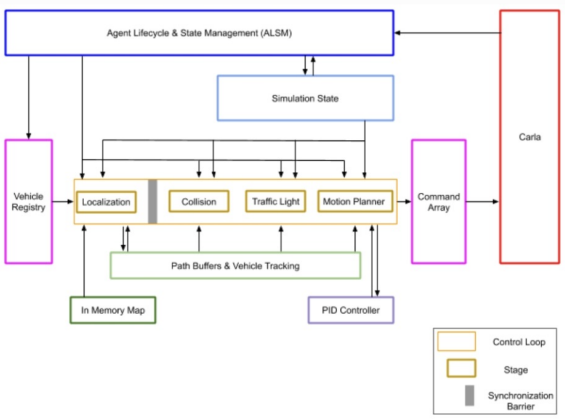

Fig. 1. The overview of Implementation

II. Related Work

In this project, our model is derived from focusing on two models Xception and ViT(Vison Transforms). In this section we plan on explaining why we chose these models to train our model for this project.

A. Xception

The Xception model, introduced by François Chollet in 2016, represents a significant evolution in convolutional neural network (CNN) architecture. Its name, short for "Extreme Inception," reflects its foundation in the Inception architecture but with critical improvements. The key innovation in

Xception is the replacement of the traditional Inception modules with depthwise separable convolutions, enhancing both efficiency and performance.

- **Depthwise separable convolutions** decompose the standard convolution operation into two simpler operations: depthwise convolutions and pointwise convolutions. This separation significantly reduces computational complexity and the number of parameters, leading to faster and more efficient neural networks.
- **Pointwise Convolution** Following the depthwise convolution, a pointwise convolution applies a 1x1 convolution to combine the outputs from the depthwise convolution. This operation linearly combines the depthwise convolution outputs across all channels.For example, If the depthwise convolution produces three feature maps, a pointwise convolution with filters will combine these feature maps to produce output channels.

$$StandardConvolutionCost = D_i \times D_o \times K \times K \times H \times W) \quad (1)$$

Depthwise Separable Convolution Cost

$$= (D_i \times K \times K \times H \times W) + (D_i \times D_o \times H \times W) \quad (2)$$

Where, Di is the number of input channels. Do is the number of output channels. K is the kernel size. H and W are the height and width of the feature maps.

B. ViT(Vision Transformers)

Vision Transformers (ViTs), introduced by Dosovitskiy et al. in 2020, adapt the transformer architecture from natural language processing to computer vision tasks. Unlike traditional convolutional neural networks (CNNs), which process images through a series of localized operations, ViTs divide an image into fixed-size patches and treat these patches as sequences of tokens. The transformer processes these sequences using self-attention mechanisms, allowing it to capture global context and long-range dependencies within the image. This holistic approach often results in superior performance on image recognition tasks, especially when large datasets are available.

The ViT architecture consists of a patch embedding layer that converts image patches into embedding vectors, followed by multiple layers of transformer encoders. These encoders utilize multi-head self-attention and feed-forward networks to process and refine the image representations. A special classification token (CLS) is included in the input sequence, and its final state is used for classification tasks. ViTs offer advantages such as enhanced scene understanding and robustness to varied input data, making them particularly suitable for applications in autonomous driving where comprehensive environmental perception is crucial.

Despite their advantages, ViTs face challenges such as high data and computational requirements. They typically need large amounts of training data and powerful hardware for optimal performance. However, their ability to integrate global context and handle diverse input types positions ViTs as a promising technology in autonomous driving and other computer vision applications. As research progresses, ViTs are expected to become more efficient and widely adopted, driving advancements in autonomous vehicle perception and decision-making systems.

III. Methodology

A. Data Collection and Augmentation

The data collection for this project was conducted using the CARLA simulator, specifically employing its autopilot function to ensure a consistent and scalable dataset. Using an autopilot instead of human drivers provides uniformity in driving patterns and helps in collecting large amounts of data efficiently. The primary objective is to gather a diverse dataset that can train the model to generalize well across various driving scenarios.

To further enhance the model's robustness and generalization capabilities, several data augmentation techniques are applied:

- **Salt-Pepper Noise:** This technique involves adding random noise to the images, simulating the sensor imperfections and noise that can occur in real-world conditions. This helps the model learn to handle noisy inputs effectively.
- **Gaussian Blur:**Blurring the images mimics various weather conditions (such as fog or rain) and camera focus issues. This augmentation ensures that the model can still perform well even when the input images are not perfectly clear.
- **Darkening:**Adjusting the brightness of the images to simulate different lighting conditions, such as driving at night or in low-light environments. This trains the model to be adaptable to a wide range of lighting conditions.
- **Cropping:**Randomly cropping parts of the images to simulate different viewpoints and focus areas. This technique helps the model learn to identify relevant features even when parts of the scene are missing or occluded.

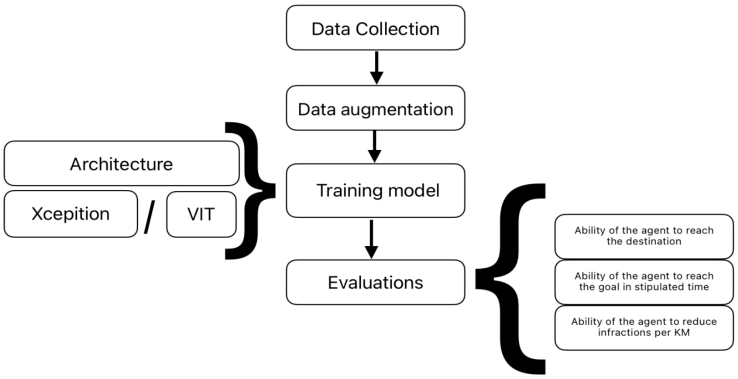

Fig. 2. Illustrates the pipeline of our methodology

B. Model Architecture

The proposed model architecture combines two advanced neural network models: Xception and Vision Transformers

(ViT), each chosen for their unique strengths in handling complex visual data.

Xception: Intuition: The Xception model is selected for its robustness to varied inputs and its efficiency in training and inference. Xception uses depthwise separable convolutions, which significantly reduce the number of parameters and computational load compared to traditional convolutional networks. This makes the model faster and more efficient while maintaining high accuracy. Application: In the context of autonomous driving, Xception's ability to handle diverse input conditions—such as different lighting, weather, and perspectives—makes it highly suitable for real-time applications. Its efficient computation allows for faster decision-making, which is critical for the responsive control of autonomous vehicles.

Vision Transformers (ViT): Intuition: Vision Transformers are employed for their exceptional capability to capture the global context of images. Unlike convolutional networks that process local regions of the image hierarchically, ViTs treat the entire image as a sequence of patches and process them using self-attention mechanisms. This allows ViTs to model long-range dependencies and understand the overall structure and relationships within the image.

Application: For autonomous driving, the global context understanding provided by ViTs enhances the vehicle's ability to interpret complex driving scenarios. This includes understanding the spatial relationships between different objects on the road, predicting the behavior of other vehicles and pedestrians, and making informed navigation decisions. The hybrid architecture that combines Xception and Vision Transformers aims to leverage the strengths of both models. The depthwise separable convolutions in Xception provide efficient and robust feature extraction, while the self-attention mechanisms in ViTs offer a comprehensive understanding of the global context. This combination is designed to improve the overall performance and reliability of the autonomous driving system, making it more adept at handling a wide range of real-world driving condition.

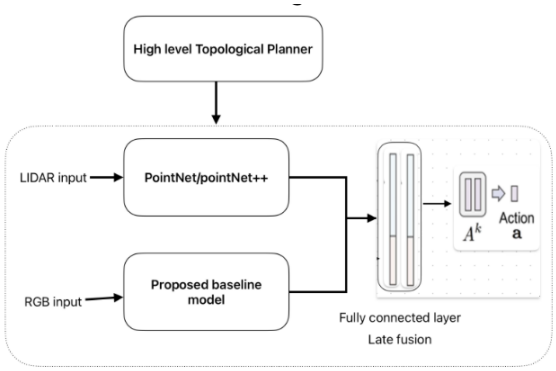

Fig. 3. architecture of the model

C. Evaluations

The evaluation phase of this project is crucial to measure the effectiveness and performance of the developed autonomous driving model. The evaluations are designed to test various aspects of the model's capabilities in real-world driving scenarios. The key metrics for evaluation include the ability to reach destinations, adherence to stipulated time periods, and minimizing infractions per kilometer driven.

Ability to Reach the Destination One of the primary evaluation metrics is the agent's ability to navigate from a starting point to a designated destination. This test assesses the overall navigation accuracy of the model. The model must demonstrate that it can consistently reach the target location without human intervention, reflecting its capability to follow the planned route effectively.

Ability to Reach the Goal in Stipulated Time The second metric evaluates whether the autonomous agent can reach its goal within a predefined time frame. This is crucial for real-world applications where timely arrivals are essential, such as in delivery services or public transportation. The stipulated time periods are determined using benchmarks set by the general autopilot system in CARLA, ensuring that the model's performance is compared against a known standard.

Ability to Reduce Infractions per Kilometer The third evaluation metric focuses on safety, specifically the model's ability to minimize infractions per kilometer driven. Infractions include: Collisions with Other Vehicles: Assessing the model's ability to avoid accidents with moving vehicles. Collisions with Pedestrians: Ensuring the model can detect and avoid pedestrians effectively. Collisions with Static Objects: Testing the model's ability to navigate around stationary obstacles like parked cars or roadblocks. Lane Breaks: Monitoring whether the vehicle stays within its designated lane, which is crucial for maintaining traffic rules and safety. Off-road Driving: Ensuring the vehicle does not veer off the road, which could indicate a failure in navigation or control. These evaluations provide a comprehensive assessment of the model's performance in realistic driving conditions. Early results indicate promising progress, with the vehicle being able to navigate in a straight line without infractions. However, further work is needed to address more complex scenarios involving turns, obstacles, and dynamic elements in the environment

IV. Results

The initial phase of our project focused on creating a dataset and customizing a pretrained model to meet the specific needs of our autonomous driving system. Our primary achievements during this phase include the following:

Dataset Creation: We collected a small dataset of 15000+ frames using the CARLA simulator. These frames represent 10 different vehicles, each annotated with relevant measurements such as speed, steering angle, and position. The dataset is in a compressed format to facilitate efficient storage and processing

Model Customization and Training:

We used a pretrained Xception model, which was customized to generate control outputs suitable for autonomous

Fig. 4. Xception in action

driving tasks. The customized model was then trained on a subset of our dataset. This training phase focused on teaching the model to handle basic control tasks such as maintaining a straight path Agent Development and Pipeline Integration:

We extended the capabilities of the Basic Agent in the CARLA simulator to form a new agent. This new agent integrates our customized Xception model into the autonomous driving pipeline. The entire process, from data collection and augmentation to model training and evaluation, was streamlined to ensure seamless operation Preliminary Successes: Initial tests demonstrated that our agent could move the vehicle in a straight line without any infractions. This indicates that the model can successfully interpret control commands and maintain a stable trajectory under simple conditions. However, additional work is required to handle more complex driving scenarios involving turns, obstacles, and dynamic elements such as other vehicles and pedestrians These early results are promising and lay a solid foundation for further development. The next steps involve expanding the dataset, refining the model to handle more complex tasks, and performing comprehensive evaluations to ensure robustness and reliability under various driving conditions.

Our initial attempts to train and evaluate the autonomous driving models yielded mixed results. While we observed some positive outcomes, several challenges remain that need to be addressed to achieve successful and reliable performance. The table below summarizes the results of our evaluations for the Basic Agent, Xception model, and Vision Transformer (ViT) model:

Model	Basic Agent	Xception	ViT
Training frames	–	15000*	1500*
Epochs	–	10	10
Evaluations			
1. Destination	Reached	Not Reached	Not Reached
2. Time	39.8s	50s	50s
3. Infraction/meter	0	49	20
Distance Traveled	160 Carla Units	30 Carla Units	40.63 Carla Units

Fig. 5. Comparisons between models

Training Frames and Epochs: The Xception model was trained on 15,000 frames over 10 epochs, while the ViT model was trained on 1,500 frames over 10 epochs. The Basic Agent did not undergo additional training as it serves as a baseline for comparison.

Destination Reaching Capability: The Basic Agent successfully reached the destination, demonstrating its ability to navigate to a target location. In contrast, both the Xception and ViT models failed to reach the destination within the tested scenarios, indicating that further tuning and training are necessary to improve their navigation capabilities.

Time to Reach Destination: The Basic Agent reached the destination in 39.8 seconds, which is faster than the Xception and ViT models, both of which took 50 seconds. This suggests that the Basic Agent currently has a more efficient navigation strategy under the given conditions.

Infractions per Meter: The Basic Agent had zero infractions per meter, showing its ability to navigate without collisions or other errors. The Xception model, however, had 49 infractions per meter, highlighting significant challenges in maintaining safe navigation. The ViT model performed better than the Xception model with 20 infractions per meter but still needs improvement to be considered safe for real-world deployment.

Distance Traveled: The Basic Agent traveled a total distance of 160 Carla Units, far surpassing the distances traveled by the Xception (30 Carla Units) and ViT (40.63 Carla Units) models. This indicates that the Basic Agent is more reliable over longer distances, whereas the other models struggled to

maintain consistent performance. Despite the initial efforts and the potential shown by advanced models like Xception and Vision Transformers, our current implementations did not achieve successful results. The Xception and ViT models were unable to reach the destination and exhibited higher infraction rates compared to the Basic Agent. These findings highlight the need for further refinement in data augmentation, model training, and evaluation processes. Future work will focus on addressing these challenges to improve the robustness and reliability of our autonomous driving system.

Conclusion and Remarks

In this project, we set out to develop a robust end-to-end autonomous driving system using advanced neural network models, specifically Xception and Vision Transformers (ViTs). Our goal was to leverage the strengths of these models to create a reliable and efficient autonomous driving architecture capable of handling diverse and complex driving scenarios.

We began by collecting and augmenting a dataset using the CARLA simulator, ensuring a consistent and scalable foundation for training our models. The data was processed with various augmentation techniques to enhance model generalization. Our customized Xception model and ViT architecture were integrated into the autonomous driving pipeline, and preliminary evaluations were conducted to assess their performance.

The results of our evaluations were mixed. While the Basic Agent successfully reached the destination and maintained a high level of safety with zero infractions, both the Xception and ViT models faced challenges. The advanced models did not reach the destination and exhibited higher infraction rates, indicating the need for further refinement and more extensive training.

Despite these setbacks, the project demonstrated the potential of using Xception and ViT models in autonomous driving applications. The Xception model's efficiency and robustness, combined with the ViT's ability to capture global context, offer promising avenues for future development. However, due to limitations in resources and time, we were unable to fully realize the capabilities of these models within the project timeframe.

Given adequate resources and additional time, we are confident that the models could be refined and optimized to achieve the desired performance. Future efforts will focus on enhancing the training processes, expanding the dataset, and conducting comprehensive evaluations to ensure the models' robustness and reliability. With these improvements, the project holds significant potential to contribute to the advancement of autonomous driving technology, paving the way for safer and more efficient autonomous vehicles. To optimize data loading and processing during model training, we employed pinned memory, batched batch execution, and multiple data loading workers. Pinned Memory: page-locked memory, allows faster data transfer between the CPU and GPU, reducing overhead and improving efficiency, especially with large datasets. Batched Batch Execution We used batched batch execution to process data in mini-batches, enhancing GPU utilization and stabilizing training dynamics. This method speeds up training and improves convergence by averaging gradients over multiple samples. Number of Workers: We configured multiple worker threads for parallel data loading, significantly reducing the waiting time for data to be transferred to the GPU. This parallelism ensures full GPU utilization and enhances overall training efficiency.These optimizations enabled us to maximize our training pipeline's performance within the constraints of our computational resources.

References

[1] https://carla.readthedocs.io/en/0.9.15/tuto$_{G}bounding_{b}oxeshttps$: $//brax.gg/carla-agent-end-to-end-imitation-learning$

[2] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N. (2020).

[3] Chollet, F. (2016). Xception: Deep Learning with Depthwise Separable Convolutions. ArXiv. /abs/1610.02357