Autonomous Driving Machine Learning Model

[Your Name]
[Your Institution]
[Your Email]

***Abstract*—This report provides an update on the progress of a machine learning project focused on autonomous driving. The goal is to develop a model that can detect lanes and objects in real-time, allowing vehicles to navigate without human intervention. The project uses the BDD100K Dataset, which contains labeled images of driving scenarios. So far, the work includes data preprocessing, annotation adjustments, and initial model setup. The next steps involve training the model, testing its performance, and evaluating its accuracy.**

***Index Terms*—Autonomous Driving, Machine Learning, Lane Detection, Object Detection, BDD100K Dataset**

I. Introduction

Autonomous driving technology aims to reduce human involvement in driving by using machine learning algorithms. A key challenge is enabling vehicles to detect lanes and recognize objects like cars, pedestrians, and obstacles. This project focuses on building a machine learning model to address these challenges, specifically for lane and object detection. The model is being trained using the BDD100K Dataset, which provides labeled images with annotations for objects and lane markings. The goal is to create a model that can accurately identify lanes and objects, which is essential for autonomous navigation in real-world environments.

II. Dataset

The BDD100K Dataset (Berkeley DeepDrive 100K) is used in this project. It contains a large collection of images from driving scenarios, along with annotations for objects and lane markings. The dataset is divided into three parts:

- **Training Set:** Used to train the machine learning model.
- **Validation Set:** Used to check the model's performance during training.
- **Test Set:** Used for final evaluation of the model.

Each image in the dataset comes with annotations stored in JSON files. These annotations include polygon points for lane markings and bounding boxes for objects like cars and pedestrians.

III. Methodology

A. Data Preprocessing

Data preprocessing is an important step to ensure the model receives consistent and properly formatted input. The following steps were applied:

- **Image Resizing:** All images were resized to 224x224 pixels, a common size for deep learning models.
- **Annotation Adjustment:** Since the images were resized, the annotation coordinates were scaled to match the new dimensions.

B. Model Development

The project uses OpenCV for image processing tasks, such as drawing bounding boxes and polygons. The Matplotlib library is used to visualize the results. Currently, the focus is on data preparation and visualizing lane and object annotations on images.

C. GitHub Repository

The project is stored on GitHub, which includes:

- `main.py`: The main script to run the project.
- `requirements.txt`: List of required libraries.
- `data/`: Folder containing images and annotations.

- `support/`: Utility files for data loading, processing, and visualization.
- `others/`: Contains reports, presentations, and demo videos.

The GitHub link is available for further review and updates: [GitHubRepository].

IV. Current Progress

As of now, the following milestones have been achieved:

- **Data Loading and Preprocessing:** The images and annotations have been successfully loaded. Images are resized to 224x224, and annotations are adjusted to match the new dimensions.
- **Visualization:** A function was created to display images along with their annotations, such as lane markings and object bounding boxes.
- **Initial Model Setup:** The basic structure for the machine learning model has been set up, focusing on image processing and annotation handling.

V. Challenges and Solutions

A few challenges were encountered during the project:

- **Annotation Scaling:** When resizing the images, the annotation points needed to be scaled to maintain accuracy. This was resolved by using the resize ratio to adjust the coordinates.
- **Visualization Issues:** Initially, the annotated polygons and bounding boxes did not display correctly. This was fixed by converting the images from BGR to RGB format for proper visualization in Matplotlib.

VI. Next Steps

The next steps for the project include:

- **Model Training:** The model will be trained using the BDD100K dataset to detect lanes and objects in the images.
- **Testing and Evaluation:** After training, the model will be tested on the validation set, and its performance will be evaluated based on accuracy and speed.
- **Further Enhancements:** Future work will involve improving the model by experimenting with different architectures and hyperparameters.

VII. Conclusion

The autonomous driving machine learning model is progressing well. The initial stages focused on data preprocessing, annotation handling, and model setup. The BDD100K Dataset has been successfully integrated into the project, and image processing tasks have been handled using OpenCV and Matplotlib. The next steps involve training the model and testing its performance for accurate lane and object detection.

References

[1] Y. Yu et al., "BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2636–2645, 2020.

[2] F. R. Cordeiro, H. G. Nascimento, and D. Menotti, "Deep Learning for Road Object Detection and Classification in Autonomous Driving: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 5, pp. 4766–4784, 2023.

[3] G. Neuhold, T. Ollmann, S. Rota Bulò, and P. Kontschieder, "The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 4990–4999, 2017.

[4] OpenCV, "Open Source Computer Vision Library," [Online]. Available: https://opencv.org. [Accessed: Feb. 2025].

[5] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, pp. 8024–8035, 2019.

[6] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.