Traffic sign detection

Project ID: PRAICP-1004-RainfallTS

Project Team ID: PTID-AI-FEB-25-1079

Traffic sign detection plays a crucial role in modern transportation systems,

Using a deep learning approach, we trained an EfficientNet-based convolutional neural network (CNN) for high accuracy and efficient performance.

In this project we build a Machine Learning Model based on provided dataset.

This project completed while doing DataMites Internship.

Submitted By:

Madhuri Mane

Hoor Fatima Hussain

Priyanka Patil

Naveen Kanna

Traffic sign detection Analysis Report

1.Introduction

This project focuses on building an image classification model capable of detecting and classifying traffic signs into 43 distinct categories. Using a deep learning approach, we trained an EfficientNet-based convolutional neural network (CNN) for high accuracy and efficient performance.

2. BUSINESS CASE

Develop a Machine Learning (ML) model to predict the traffic sign in the given frame

3. OBJECTIVES

- Develop a high-performance traffic sign detection model.

- Ensure the model generalizes well on external data.

4. PROJECT GOAL

1. Data understanding and preprocessing.

2. ML model to Traffic sign detection.

5. DATA OVERVIEW

Dataset Source: Provided as a folder named traffic_data

Number of Classes: 43

Training Data: Separate training directory with labeled subfolders

Validation Data: Separate validation directory

Test Data: Test directory with metadata provided in CSV files

Image Resolution: All images resized to (224, 224)

6. METHODOLOGY

❖ Data Extraction:

▪ Data extraction is a fundamental process in data management, and it plays a critical role in preparing data for analysis, machine learning.

❖ Data preprocessing:

▪ Data Cleaning: Cleaning data from dataset. ▪ Handling Missing Values: Removal of rows or columns with excessive missing values.

❖ Data Transformation:

▪ Normalization and Scaling: Rescaling numerical data to a specific range (e.g., 0 to 1) or standardizing it to have a mean of 0 and a standard deviation of 1.

❖ Models:

<u>Chosen Model:</u>

EfficientNet

<u>Why EfficientNet</u>:

Lightweight and efficient with strong performance on image classification task

Pretrained on ImageNet, allowing transfer learning for faster training.

<u>Architecture:</u>

<u>Input Shape:</u> (224, 224, 3)

<u>Pretrained Weights:</u> ImageNet

<u>Dense Output Layer:</u> 43 neurons with softmax activation

<u>Loss Function:</u> Categorical Crossentropy

<u>Optimizer:</u> Adam

<u>Evaluation Metric</u>: Accuracy

❖ Accuracy Plots

Training and Validation Accuracy Plot:

This shows how well your model learns over epochs. A rising training accuracy with a stable or improving validation accuracy means your model is generalizing well.

Training and Validation Loss Plot:

This helps track how your model minimizes the error. A decreasing loss for both training and validation sets indicates better performance.

Confusion Matrix:

Visualizes the model's predictions compared to actual labels, showing where it gets confused between traffic sign classes.

Classification Report (Optional as a Heatmap):

Displays precision, recall, and F1-score for each class, helping you identify which classes the model struggles with.

7. CHALLENGES

Low Initial Accuracy:

- Switched from custom CNN to EfficientNet for better performance.
- Applied data augmentation to prevent overfitting and improve generalization.

Class Imbalance:

- Verified dataset balance and adjusted augmentation strategies accordingly.

8. SUMMERY

Model: EfficientNet (pretrained) for traffic sign classification.

Dataset: 43 traffic sign classes, images resized to (240, 240).

Data Preprocessing: Used ImageDataGenerator() for augmentation (like rotation, zoom, and flipping).

Training: Split into training and validation sets; tuned for better performance.

Performance: Achieved good accuracy and minimized loss.

Model Saving: Final trained model stored in Google Drive’s "traffic models" folder.

9. Conclusion

The traffic sign detection model using EfficientNet achieved excellent performance with 97% training accuracy and 89% validation accuracy. The deployment through Flask and Google Colab allowed easy testing via a web interface. This project demonstrates the effectiveness of transfer learning for traffic sign classification and sets the foundation for real-world applications in autonomous driving systems.