ARTIFICIAL INTELLIGENCE PROJECT REPORT

Roll: 2023BCD0031

Name : Kamma Sainishitha

1. Title of the Project

"GERMAN TRAFFIC SIGN RECOGNITION AND CLASSIFICATION"

2. Abstract

Traffic sign recognition is a crucial component of modern intelligent transportation systems. This project focuses on classifying German traffic signs using a Convolutional Neural Network (CNN) model .The dataset contains 43 traffic sign categories with significant variation in appearance, size, and lighting conditions. The project includes data loading, visualization, preprocessing, augmentation, model training, and prediction on new images.

The CNN model achieves high accuracy after augmentation and training, demonstrating that it can effectively identify traffic signs in real-world scenarios. This project aims to show how learning can assist in improving road safety, autonomous driving, and advanced driver-assistance systems .

3. Introduction

Traffic signs convey essential information to drivers and autonomous vehicles. Recognizing these signs automatically helps in:

- Avoiding road accidents
- Assisting driver awareness
- Supporting self-driving car navigation

Traditional machine learning methods require handcrafted features, whereas CNNs automatically learn spatial patterns from images.

This project uses the German Traffic Sign Recognition Benchmark (GTSRB) dataset,Since due to due to high server traffic the dataset is temporarily removed from the website so I have uploaded it in a google drive to access it publicly

Dataset link :

https://drive.google.com/drive/folders/1xaX57q3E-KuGvOU4qerh4r3BXp4GJsO-?usp=sharing

4. Objectives

1. To load and preprocess the German Traffic Signs dataset.
2. To visualize the dataset and understand class distribution.
3. To apply data augmentation for improving class balance and accuracy.
4. To design and train a CNN model using TensorFlow.
5. To evaluate the model's accuracy and performance.
6. To test the trained model on new images.

5. Dataset Description

- Total samples: 4410
- Number of classes: 43
- Format: PNG images
- Labels provided through a CSV file (image name, label, bounding box, width, height)

Examples of traffic sign classes include:
Speed limits, Stop, No Entry, Pedestrians, Traffic Signals, Roundabout, Children Crossing, etc.

Dataset issues identified:

- Unbalanced classes
- Some classes contain very few samples
- Images vary greatly in size and appearance

To address these issues, data augmentation was applied.

6. Methodology

6.1 Data Loading

- Loaded the metadata from train_dataset.csv.
- Extracted file names, labels, bounding box dimensions, and image shapes.

6.2 Data Visualization

- Explored dataset class distribution using bar plots.
- Displayed example images for each class to observe variations.

- Identified underrepresented classes (e.g., classes with only 30 images).

7. Data Preprocessing

7.1 Data Augmentation

To improve the model's robustness and balance the dataset:

- **Rotation** (+10°, -10°)
- **Flipping** (for symmetrical or reversible signs)
- **Scaling and translation** (where applicable)

After augmentation:

- Training samples increased from 4410 → 13230.

7.2 Normalization

Pixel values normalized as:

$$(x - 128)/128$$

This centers data around 0 and speeds up training.

7.3 Conversion to Grayscale

Converted RGB images to grayscale:

- Reduces computational cost
- No information loss for sign shapes
- Final image shape: 32 × 32 × 1

8. Model Architecture (LeNet CNN)

The CNN consists of:

1. Input Layer – 32×32×1 grayscale images
2. Conv Layer 1 – 16 filters, 3×3, ReLU activation
3. Max-Pooling – 2×2
4. Conv Layer 2 – 64 filters, 3×3, ReLU activation

5. Max-Pooling – 2×2
6. Conv Layer 3 – 256 filters, 3×3, ReLU activation
7. Max-Pooling – 2×2
8. Flatten Layer
9. Fully Connected Layer – 120 neurons, ReLU
10. Dropout (keep_prob = 0.7)
11. Output Layer – 43 classes (softmax applied during inference)

Optimizer: Adam
Learning Rate: 0.0005
Batch Size: 128
Epochs: 50

9. Training Process

- Training data shuffled each epoch for unbiased learning.
- Loss function: **softmax cross entropy**.
- Optimizer: **Adam** adjusts weights during backpropagation.
- Accuracy improved steadily across epochs.

Training Accuracy Progression

Example values from the log:

- Epoch 1: 31%
- Epoch 10: 96%
- Epoch 25: 99%
- Epoch 50: 99.8%

The model showed excellent convergence after augmentation.

10. Evaluation

Evaluation was performed after each epoch using a batch-wise accuracy computation.
The final training accuracy reached **99–100%**, indicating strong learning ability.

A TensorFlow Saver() was used to store the trained model as lenet.

11. Testing on New Images

The model was used to classify user-selected images via a file dialog.
Steps include:

1. Load image (PNG/JPG)
2. Perform the data augmentation.
3. Convert to grayscale
4. Resize to 32×32
5. Normalize
6. Predict class using the restored model

Example Predictions

- Stop Sign → 99.64%
- General Caution → 99.96%
- No Entry → 99.99%

The model showed robust generalization on unseen test images.

12. Results

- Highest training accuracy: **~99.8%**
- Strong classification even on rotated, dim, and real-world images.
- Data augmentation significantly improved performance.
- LeNet architecture proved effective for traffic sign classification.

13. Applications

- Autonomous vehicles
- Driver Assistance Systems
- Road safety monitoring
- Smart traffic systems
- Image-based navigation

14. Conclusion

This mini project demonstrates the implementation of a CNN-based traffic sign classifier using the LeNet architecture. By combining data preprocessing, augmentation, and deep learning, the model achieves very high accuracy and handles real-world images effectively.

It highlights the potential of deep learning models in automotive safety and intelligent transportation systems.

15. Future Scope

1. Using color images instead of grayscale
2. Employing advanced architectures.
3. Training on larger datasets (GTSRB full dataset)
4. Deploying on mobile/embedded systems (TensorFlow Lite)

---

16. References

1. Sermanet, Pierre, and Yann LeCun. “Traffic Sign Recognition with Multi-Scale Convolutional Networks.”
2. German Traffic Sign Recognition Benchmark (GTSRB).
3. TensorFlow Documentation.
4. OpenCV Documentation.