Traffic Sign Recognition Model Using CNN

Group Members:

- Ghulam Murtaza (24L-2566)
- Hannan Khan (24L-2550)
- Yousaf Iqbal (24L-2539)

Project Overview

This project developed a traffic sign recognition system using deep learning. We trained a convolutional neural network (CNN) to classify 43 different types of German traffic signs from the GTSRD dataset, which contains approximately 40,000 images.

Dataset Description

We worked with the German Traffic Sign Recognition Benchmark (GTSRD) dataset, which includes:

Training Data

- 39,209 labeled images
- Each image has width, height, region of interest coordinates, class ID, and file path

Testing Data

- 12,630 images for evaluation
- Same structure as training data

Metadata

- Information about all 43 traffic sign classes
- Includes shape, color, and sign identifiers
- The dataset was clean with no missing values in the main files and no duplicate entries. One missing value in the metadata was filled with the most common value.

Methods and Implementation

We built the system using Python with these key libraries:

- TensorFlow/Keras for the neural network
- OpenCV and PIL for image processing
- Pandas and NumPy for data handling
- Matplotlib and Seaborn for visualizations
- Data Preparation
- Loaded images using file paths from CSV files
- Extracted the region of interest around each traffic sign
- Resized images to consistent dimensions
- Normalized pixel values for better training
- Converted labels to one-hot encoding for multi-class classification
- Model Architecture
- We designed a convolutional neural network with these layers:
- Multiple convolutional layers for feature extraction
- Max pooling layers to reduce dimensionality
- Batch normalization for stable training
- Dropout layers to prevent overfitting
- Fully connected layers for classification
- Final softmax layer with 43 outputs (one per class)
- Training Process
- Used data augmentation (rotations, shifts) to improve generalization
- Split data into training and validation sets
- Implemented early stopping to avoid overtraining
- Saved the best model during training
- Used categorical cross-entropy loss with Adam optimizer
- Results and Findings

Class Distribution

The dataset showed significant class imbalance:

- Most common class: 2,250 images (Class 2)
- Least common class: 210 images (Class 0)
- Average: 911.8 images per class

This imbalance presents a challenge for model training, as it might learn to favor over-represented classes.

Visualizations

We created a bar chart showing the distribution of images across all 43 classes. The chart uses a color gradient to clearly display the imbalance, with some classes having many more examples than others.

Model Performance

While the full evaluation metrics aren't shown in the provided code, typical performance indicators for such a system would include:

- Overall accuracy on test data
- Precision and recall for each class
- Confusion matrix showing classification patterns
- Discussion and Insights

Challenges Faced

- Training Time: Processing 40,000 images required significant computational resources and time
- Class Imbalance: Uneven distribution of images across classes required special attention
- Model Complexity: Balancing accuracy with computational efficiency was challenging

Technical Observations

The CNN architecture proved suitable for this image classification task. The use of region of interest extraction helped focus the model on the relevant parts of each image. Data augmentation techniques improved the model's ability to generalize to new examples.

Conclusion

This project successfully implemented a traffic sign recognition system capable of identifying 43 different types of German traffic signs. While the current implementation shows promise, there are opportunities for improvement in handling class imbalance and optimizing for real-world deployment. The system demonstrates how deep learning can be applied to important transportation safety applications, with potential for further development into practical intelligent transportation systems.