German Traffic Sign Recognition Using Deep Learning with PyTorch

Heba Alsharif
Fanshawe College, London, Ontario, Canada
h_alsharif@fanshaweonline.ca
Professor: Mohammed Yousefhussien

Abstract

This paper presents a deep learning–based traffic sign recognition system using the German Traffic Sign Recognition Benchmark (GTSRB). The project implements a complete PyTorch pipeline including preprocessing, data augmentation, a custom CNN classifier, transfer learning using ResNet-18, a denoising autoencoder, and a DCGAN for synthetic image generation. The final model achieved 99.4% accuracy across 43 traffic sign classes and demonstrated strong robustness to noise, lighting variation, and rotation. The results show the effectiveness of combining augmentation, regularization, and modern architecture for high-precision traffic sign classification.

Introduction

Traffic sign recognition plays a critical role in modern intelligent transportation systems, especially in autonomous driving and advanced driver assistance systems (ADAS). Reliable recognition of road signs improves safety, supports real-time navigation, and reduces human error.

However, real-world traffic sign images introduce significant challenges. Variations in lighting, weather conditions, motion blur, occlusion, camera angle changes, and background clutter can severely degrade model performance. These challenges require robust computer vision models capable of generalizing well to unseen conditions.

Deep learning, particularly Convolutional Neural Networks (CNNs), has transformed image classification tasks due to their ability to automatically learn hierarchical features. The GTSRB dataset provides a comprehensive and realistic benchmark for evaluating deep learning methods in traffic sign recognition. This project explores an end-to-end PyTorch-based solution designed to handle real-world complexity using both custom and pretrained deep learning architectures EASE OF USE

DATASET

The GTSRB dataset contains more than 50,000 training images and 12,000 test images, covering 43 distinct traffic sign classes including speed limits, warnings, prohibitions, and mandatory signs. Images vary significantly in brightness, contrast, scale, rotation, shadows, occlusions, and background noise—making the dataset suitable for evaluating robust classification systems.

Data was split as follows:

- Training: 80%
- Validation: 10%
- Testing: 10%

To ensure high generalization, each subset preserved class distribution. The dataset's natural variability plays an essential role in training models that can perform reliably under real-world conditions.

METHODOLOGY

A. Preprocessing

All images were resized to 64×64 pixels to maintain uniform input size and computational efficiency. Pixel values were normalized using standard ImageNet mean and standard deviation values to stabilize training and accelerate convergence. Corrupted or unreadable samples were ignored automatically during loading.

B. Data Augmentation

Data augmentation was used extensively to increase robustness and reduce overfitting. The training pipeline applied:

- Random rotation
- Random horizontal flipping
- Color jitter (brightness and contrast variation)

These transformations simulate real driving conditions such as camera shake, different lighting, or changing viewpoints.

C. Custom CNN Architecture

A lightweight CNN was built with the following structure:

- Three convolutional blocks (32 → 64 → 128 filters)
- ReLU activation after each convolution
- MaxPooling for spatial downsampling
- Dropout (50%) in the classifier head
- Total parameters: ~1.5 million

This custom model served as the baseline for performance comparison.

D. Transfer Learning Using ResNet-18

A pretrained ResNet-18 model was fine-tuned as a secondary approach. Early layers were frozen to retain robust ImageNet feature extraction capabilities, while the final classifier layer was replaced with a custom linear head tailored for the 43-class GTSRB dataset. Transfer learning significantly improved feature representation and performance stability

---

TRAINING STRATEGY

A. Hyperparameters

- Optimizer: Adam (learning rate = 0.001)
- Loss function: Cross-Entropy
- Batch size: 64
- Epochs: 20
- Train/validation split: 80/20

B. Regularization Techniques

- Dropout in the classifier
- Batch normalization
- Learning rate scheduler
- Early stopping (patience = 5 epochs)

These strategies minimized overfitting and ensured stable convergence.

RESULTS AND DISCUSSION

A. Model Performance

The final model achieved 99.4% test accuracy, outperforming the baseline CNN. The learning curves showed smooth convergence, confirming the effectiveness of regularization and augmentation.

B. Confusion Matrix Insights

Most classes achieved near-perfect precision and recall. Errors occurred mainly between visually similar signs, such as speed limits with similar shapes. Errors were rare and generally caused by poor lighting or partial occlusions.

C. Training Behavior

The model demonstrated:

- Healthy loss and accuracy progression
- No signs of overfitting due to augmentation and early stopping
- Stable feature learning

DENOISING AUTOENCODER

A denoising autoencoder was trained to reconstruct clean traffic sign images from noisy inputs. The encoder compressed features into a latent representation, while the decoder restored the image. Results showed successful recovery of structure and color, even under heavy noise. This component demonstrates the usefulness of generative models for preprocessing and improving dataset quality

quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)

DCGAN FOR SYNTHETIC IMAGE GENERATION

A Deep Convolutional GAN (DCGAN) was trained to produce synthetic traffic sign images. The generator learned to create visually realistic signs, while the discriminator distinguished real images from generated ones. Although early training produced blurry results, later epochs generated clearer shapes and structures. Synthetic samples can support dataset expansion and improved model generalization

---

CONCLUSION

This project successfully implemented a complete deep learning pipeline for traffic sign recognition using PyTorch. By integrating a custom CNN, transfer learning, augmentation, regularization, a denoising autoencoder, and a DCGAN, the system achieved strong accuracy and robustness.

Future work includes ONNX deployment for real-time use, extended data augmentation strategies, nighttime/poor-weather datasets, and more advanced architectures such as transformers or attention-based models

ACKNOWLEDGMENT

I would like to thank Professor Mohammed Yousefhussien for his continuous guidance, valuable feedback, and support throughout the development of this project.

Appreciation is also extended to Fanshawe College for providing the computational resources and learning environment that enabled the successful completion of this work.

REFERENCES

[1] S. Houben et al., “Detection of Traffic Signs in Real-World Images: The GTSRB Benchmark,” in Proc. Int. Joint Conf. Neural Networks (IJCNN), 2011.

[2] K. He, X. Zhang, S. Ren and J. Sun, “Deep Residual Learning for Image Recognition,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016.

[3] PyTorch Documentation, “Deep Learning Framework,” Available: https://pytorch.org/.

[4] D. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” arXiv:1312.6114, 2013.

[5] A. Radford, L. Metz and S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” arXiv:1511.06434, 2015.

FANSHAWE