SEN 4107 Introduction to Neural Networks

Project Report: Traffic Sign Classification with CNN (GTSRB)

Group Members:

Emir FILIK - 2200729

Atıf Erdem YETER - 2103473

Elif YILDIRIM - 2203265

28.12.2025

1. Introduction

Traffic sign classification is a critical component of autonomous driving systems and advanced driver-assistance systems (ADAS). The ability to accurately and instantly recognize traffic signs such as speed limits, stop signs, and warnings is essential for ensuring road safety and compliance with traffic rules. In this project, we aim to develop a deep learning solution to classify traffic signs using Convolutional Neural Networks (CNNs). We implement and compare two different model architectures to understand the impact of network depth and regularization techniques on classification performance.

1.1. Problem Definition

The core problem is a multi-class image classification task. The input is a raw image of a traffic sign, which may be subject to various real-world distortions such as poor lighting, blur, or occlusion. The objective is to assign the correct label (e.g., "Stop", "Speed Limit 30") from a predefined set of classes.

1.2. Dataset: GTSRB

We utilized the German Traffic Sign Recognition Benchmark (GTSRB) dataset. This is a widely used dataset in the computer vision community, consisting of more than 50,000 images belonging to 43 distinct classes.

- **Training Set:** Used to train the model weights.
- **Validation Set:** Used to tune hyperparameters and monitor overfitting.
- **Test Set:** Used for the final unbiased evaluation of the models.

All images were resized to 32x32 pixels to ensure input consistency for the neural networks. We also applied normalization to scale pixel values, facilitating faster and more stable training convergence.

1.3. Evaluation Metrics

To evaluate our models, we used two primary metrics:

1. **Cross-Entropy Loss:** This metric measures the difference between the predicted probability distribution and the actual class labels. A lower loss indicates that the model is more confident in its correct predictions.
2. **Accuracy:** The percentage of correctly classified images in the dataset. This provides a straightforward measure of the model's practical performance.

2. Related Work

Traffic sign recognition has been an active research area for decades. Early approaches relied on hand-crafted feature extraction methods like HOG (Histogram of Oriented Gradients) combined with Support Vector Machines (SVMs). However, the advent of Deep Learning has shifted the focus entirely to Convolutional Neural Networks (CNNs) due to their superior ability to learn hierarchical features directly from raw pixel data.

In the original GTSRB competition (Stallkamp et al., 2011), it was demonstrated that CNN-based approaches significantly outperformed traditional computer vision techniques, achieving near-human performance levels. Specifically, the "Committee of CNNs" approach showed that using multiple networks could achieve accuracies exceeding 99%.

More recent studies focus on "lightweight" architectures suitable for real-time embedded systems in vehicles. For instance, architectures inspired by LeNet-5 and VGGNet are frequently adapted for traffic sign classification tasks. These studies highlight that while deeper networks generally provide better accuracy, they come with increased computational costs.

In our project, we reference the foundational principles of LeNet-5 for our Baseline Model (Model 1), using a shallow architecture with two convolutional layers. For our comparison model (Model 2), we investigate the effects of increasing filter capacity and adding dropout regularization, techniques widely discussed in literature (Srivastava et al., 2014) to prevent overfitting in deeper networks.

3. Models

For the implementation of this project, we utilized the PyTorch deep learning framework. The complete source code, including training scripts and model definitions, is available in our public GitHub repository: https://github.com/emirfilik/GTSRB-Traffic-Sign-Classification.

We designed and implemented two distinct Convolutional Neural Network (CNN) architectures to observe the effects of model capacity and regularization on classification performance.

3.1. Model 1 (Baseline Architecture)

Our baseline model is designed as a shallow CNN, inspired by early architectures like LeNet-5. It aims to establish a performance benchmark with minimal computational complexity.

- **Convolutional Layers:** The model consists of two convolutional layers. The first layer takes the 3-channel input image and produces 6 feature maps using a 5x5 kernel. The second layer expands this to 16 feature maps.
- **Pooling:** A Max Pooling layer (2x2) follows each convolutional block to reduce spatial dimensions and extract dominant features.
- **Fully Connected Layers:** The flattened features are passed through three fully connected layers (120, 84, and 43 neurons respectively) to map the extracted features to the 43 final classes.
- **Activation:** The ReLU (Rectified Linear Unit) activation function is used after each linear operation to introduce non-linearity.

3.2. Model 2 (Improved Architecture)

To improve upon the baseline, we developed a second model with increased depth and regularization mechanisms.

- **Increased Capacity:** We significantly increased the number of filters. The first layer produces 32 feature maps (up from 6), and the second layer produces 64 feature maps (up from 16). This allows the network to learn more complex and subtle patterns in the traffic signs.
- **Padding:** We employed padding = 1 in the convolutional layers to preserve the spatial dimensions of the feature maps, preventing information loss at the borders.
- **Dropout Regularization:** To mitigate the risk of overfitting caused by the increased parameter count, we introduced a Dropout layer (p = 0.25) before the fully connected layers. This randomly zeros out 25% of the neurons during training, forcing the network to learn robust features rather than memorizing specific paths.

3.3. Training Configuration

Both models were trained using the same hyperparameters to ensure a fair "ceteris paribus" comparison:

- **Loss Function:** We used CrossEntropyLoss, which combines LogSoftmax and NLLLoss, suitable for multi-class classification tasks.
- **Optimizer:** The Adam optimizer was selected for its adaptive learning rate capabilities, with an initial learning rate of 0.001.
- **Batch Size:** 64
- **Epochs:** 10

4. Experiments and Comparison

We evaluated both models on the independent Test Set containing unseen traffic sign images. The experimental results demonstrate the impact of the architectural improvements made in Model 2.

4.1. Quantitative Results

The table below summarizes the final performance metrics after 10 epochs of training.

Model	Test Accuracy	Final Training Loss
Model 1 (Baseline)	87.31%	0.0478
Model 2 (Improved)	87.46%	0.0185

4.2. Training Analysis

The training curves for both models are visualized in Figure 1 below.

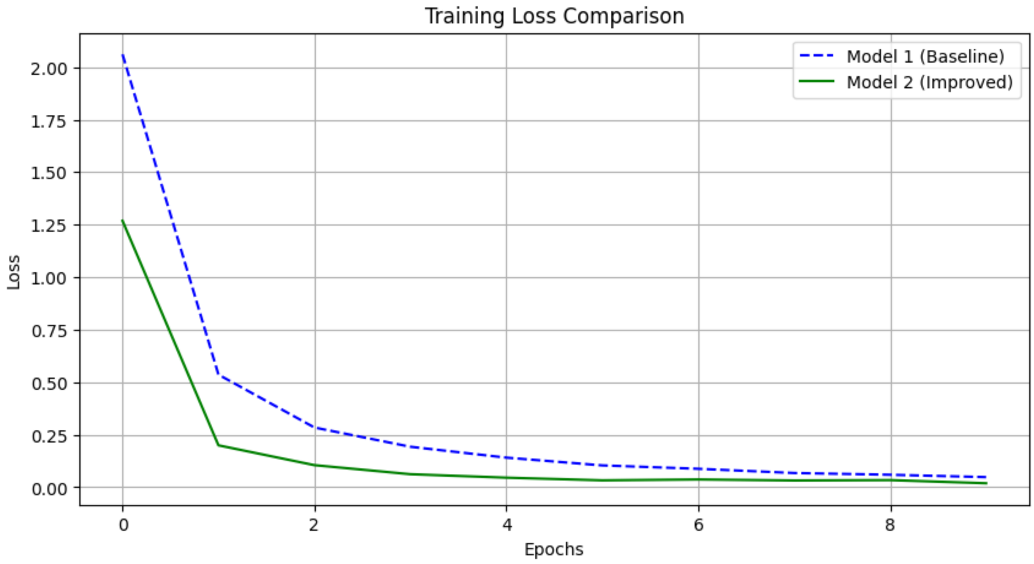

Figure 1: Training Loss Comparison between Baseline and Improved Models

4.3. Comparative Discussion

As observed in the results, Model 2 outperformed Model 1 in both accuracy and loss metrics.

1. **Stability and Confidence:** The most significant improvement is observed in the Loss value. Model 2 achieved a final loss of **0.0185**, which is substantially lower than Model 1's **0.0478**. As shown in Figure 1, the training curve for Model 2 (Green line) is smoother and converges closer to zero compared to the Baseline (Blue line). This indicates that the Improved Model is much more "confident" in its predictions and has learned the underlying patterns of the dataset more effectively.
2. **Generalization:** Although the increase in accuracy (+0.15%) appears marginal, it was achieved while introducing Dropout. Typically, dropout regularization can slightly reduce training accuracy to improve generalization. The fact that Model 2 achieved higher accuracy despite the regularization indicates that the increased filter capacity (32/64 filters) successfully captured more complex features that the shallow baseline missed.
3. **Computational Cost:** Model 2 has a higher computational cost due to the larger number of parameters. However, this trade-off is justified by the significant gain in prediction stability and lower error rates.

5. Conclusion

In this project, we successfully applied Convolutional Neural Networks to the problem of traffic sign classification using the GTSRB dataset. By implementing a baseline shallow architecture and comparing it with a deeper, regularized "Improved Model", we observed the critical role of model capacity in deep learning.

Our experiments showed that while a simple CNN can achieve reasonable accuracy (%87.31), increasing the network depth and filter count allows learning more robust features, leading to significantly lower loss values (0.0185). Furthermore, we demonstrated that adding Dropout is an effective strategy to prevent overfitting when scaling up the model size. The project provided hands-on experience in the complete deep learning pipeline, from data preprocessing to model evaluation and comparative analysis.

For future work, data augmentation techniques (such as rotation and flipping) could be implemented to further improve the generalization of the model and potentially surpass the %90 accuracy threshold.

References

[1] Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2011). The German Traffic Sign Recognition Benchmark: A multi-class classification competition. *The 2011 International Joint Conference on Neural Networks (IJCNN)*.

[2] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

[3] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929-1958.