Design, Analysis, and Evaluation of a Traffic-Sign Recognition System

Author: Amay Viswanathan Iyer (s3970066) and Basavaraj Sanshi (s3975993)

Submission Date: May 26, 2024

Introduction:

Traffic sign recognition is essential for the safety and efficiency of autonomous driving systems and advanced driver-assistance systems (ADAS). This project aims to develop and evaluate a machine learning system that can classify European road traffic signs into two categories: sign shape and sign type. The motivation is to contribute to advancements in autonomous driving technology and gain insights into the challenges of developing such models with real-world data. The project will use a modified version of the Belgium Traffic Sign Classification Benchmark, consisting of 28x28 grayscale images of road signs captured from real-world vehicles. Various supervised machine learning algorithms will be explored to develop models for accurate traffic sign classification. The project will involve data preprocessing, augmentation, and evaluation of different models. An independent evaluation phase will test the trained models on self-collected traffic sign images, simulating real-world deployment. The goal is to develop a robust and reliable traffic sign recognition system that can be integrated into autonomous driving systems, improving road safety. The insights gained from this project will benefit autonomous driving and other domains involving object recognition.

Data Preprocessing:

Before diving into model development, we performed essential data preprocessing steps to ensure the dataset was suitable for training. The `trafficsigns_dataset` was organized into directories based on shape and type, providing a structured framework for the models to learn from. We applied the following preprocessing techniques:

1. All images were processed in their consistent dimensions of 28x28 pixels (and grayscale) for CNNs and 224x224 pixels for ResNets and VGG models. This step ensures uniformity and reduces computational complexity.
2. Normalization: Pixel values were normalized to the range [0, 1], which enhances model training and convergence by keeping the values within a consistent range.

These preprocessing steps were crucial in preparing the dataset for effective model training and generalization along with Model Development.

Model Development:

We experimented with several deep learning architectures to identify the most suitable models for traffic sign shape detection and type classification. Here's an overview of the models we developed:

1. CNN Models (Baseline):
    a. Shape Detection (Model 1):
        i. 3 convolutional layers, max pooling, 2 dense layers
        ii. Captures spatial hierarchies and local patterns in images

        iii. Learns discriminative shape features through the combination of layer.

    b. Type Classification (Model 2):
        i. 4 convolutional layers, max pooling, 3 dense layers
        ii. Increased depth captures complex features for sign type classification
        iii. Additional layers enhance the capacity to learn subtle differences between sign types

2. ResNet Models:

    a. Shape Detection (Model 3):
        i. 20 layers, convolutional layers, residual blocks, dense layers
        ii. Deep architecture and residual connections facilitate learning of detailed shape features
        iii. Increased depth and skip connections capture fine-grained patterns for shape detection

    b. Hyperparameter-Tuned Shape Detection (Model 4):
        i. Similar to Model 3 (20 layers), with optimized hyperparameters
        ii. Hyperparameter tuning finds the best configuration for the specific task and dataset
        iii. Tuned hyperparameters improve convergence speed, generalization ability, and overall performance

    c. Type Classification (Model 5):
        i. 34 layers, convolutional layers, residual block.
        ii. Deeper architecture captures intricate patterns associated with different sign types
        iii. Increased depth and residual connections enhance learning of discriminative features for accurate type classification

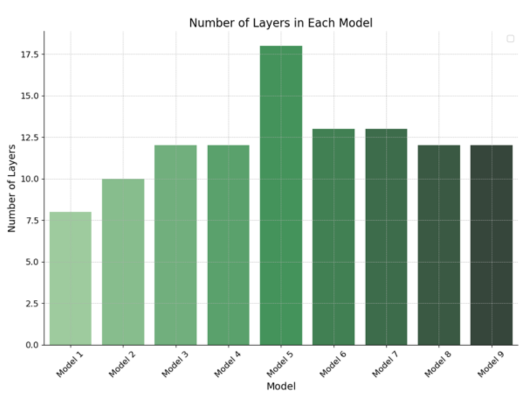
    d. Improved Type Classification (Model 6):
        i. Same as Model 5 (34 layers), with data augmentation and fine-tuned hyperparameters.
        ii. Data augmentation increases variety of training samples, while hyperparameter tuning optimizes performance
        iii. Augmented data improves robustness and generalization, while tuned hyperparameters boost classification accuracy

    e. Type Classification with Regularization (Model 7):

        i. 34 layers, L2 regularization applied to convolutional layers.
        ii. L2 regularization adds penalty term to loss function, discouraging overly complex or noise-sensitive patterns
        iii. Promotes simpler and more generalizable patterns, reducing overfitting and enhancing performance on unseen data.

3. VGG Models:
    a. Shape Detection (Model 8):
        i. 16 layers, convolutional layers, max pooling, dense layers
        ii. Deep and uniform structure allows learning hierarchical representations of shape features
        iii. Depth of the architecture enables capturing essential shape characteristics at different levels of abstraction
    b. Type Classification (Model 9):
        i. 19 layers, similar to Model 8, with increased depth for type classification
        ii. Deeper architecture provides higher level of abstraction for capturing intricate details of sign types
        iii. Increased capacity of the model, with more layers and filters, enhances ability to distinguish between sign types accurately

The graph below represents the number of layers we had for each model that we developed. As



shown, model 5 was the one with the densest architecture. It was the ResNET model that predicted type. Despite its density, it did not have a meaningful effect on the overall accuracy or f1score. Only after tuning its hyperparameters were we able to derive Model 6, which gave the best accuracy scores and f1 scores.

Hyperparameter-Tuned ResNet Models for Traffic Sign Shape and Type Detection:

In this project, we optimized the ResNet models for traffic sign shape and type detection by incorporating hyperparameter tuning. Hyperparameter tuning is a crucial step in improving the performance of machine learning models, as it involves selecting the best combination of hyperparameters that maximize the model's performance on a given task. By carefully tuning the hyperparameters, we aimed to enhance the learning capacity, generalization ability, and overall performance of the ResNet models.

Hyperparameter Tuning for Shape Detection (Model 4)

For the shape detection task, we applied hyperparameter tuning to the ResNet model (Model 4) to optimize its performance. The key hyperparameters tuned include:

1. Learning Rate
    a. The learning rate determines the step size at which the model's weights are updated during training.
    b. We experimented with different learning rates to find the optimal value that allows the model to converge efficiently and reach the best solution.
    c. A well-tuned learning rate helps the model to navigate the loss landscape effectively and find the minimum loss point.

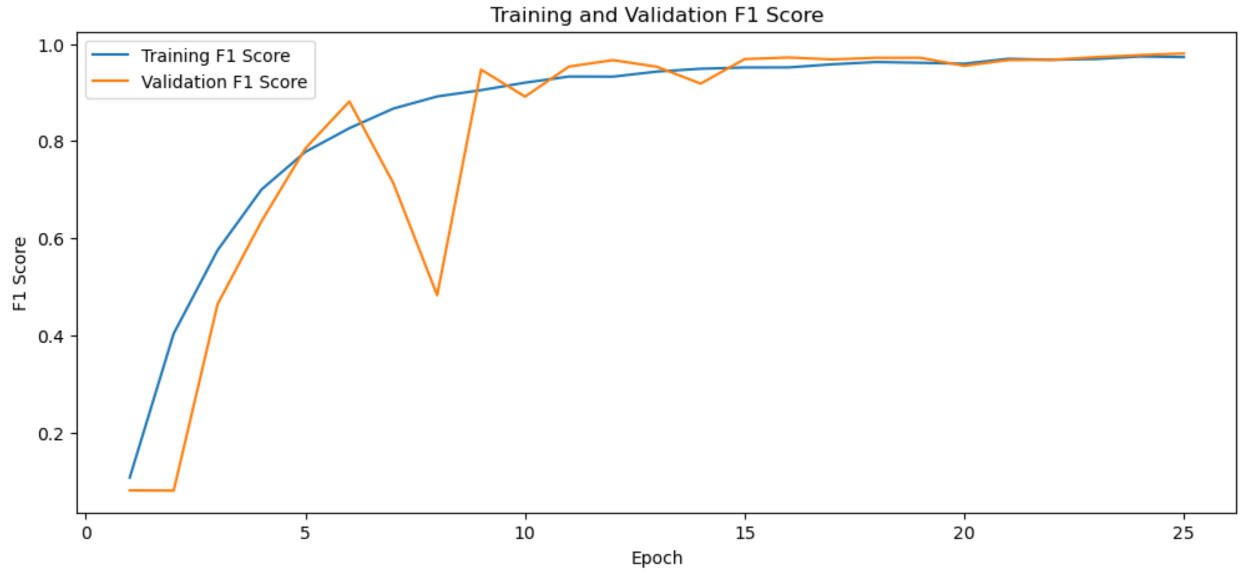
2. Regularization Strength (L2 Regularization)
    a. L2 regularization is a technique used to prevent overfitting by adding a penalty term to the loss function, discouraging the model from learning overly complex or noise-sensitive patterns.
    b. We introduced L2 regularization to the convolutional layers of the ResNet model and tuned the regularization strength hyperparameter.
    c. By finding the right balance of regularization strength, we aimed to improve the model's ability to generalize well to unseen data and reduce overfitting.

3. Number of Training Epochs:
    a. The number of training epochs determines how many times the model iterates over the entire training dataset during the learning process.
    b. We increased the number of epochs from 10 to 20 to allow the model more opportunities to learn and capture complex patterns in the shape data.
    c. However, we also employed early stopping to prevent overfitting, where training is stopped if the validation loss does not improve for a certain number of epochs.

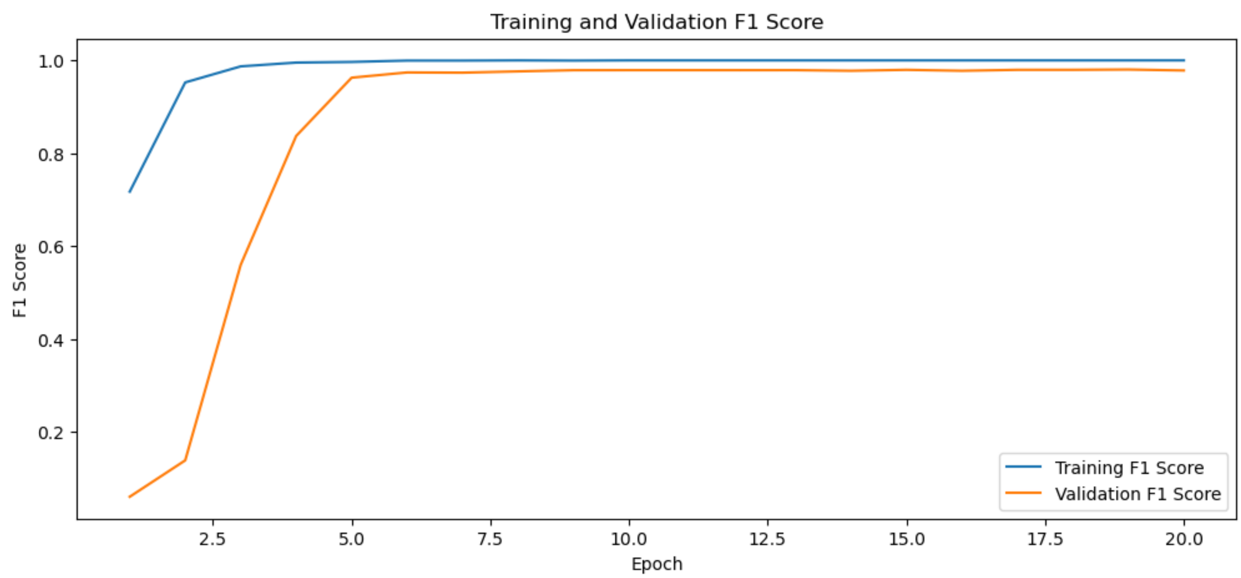
The impact of hyperparameter tuning on the shape detection model's performance is evident from the evaluation metrics:

a. The tuned ResNet model (Model 4) achieved higher accuracy and F1 scores compared to the non-tuned ResNet model (Model 3).
b. The increased depth and optimized hyperparameters enabled the model to learn more discriminative features and generalize better to unseen shape data.

Hyperparameter Tuning and Improvements for Type Classification (Model 6)
Before hyperparameter tuning:



After hyperparameter tuning:



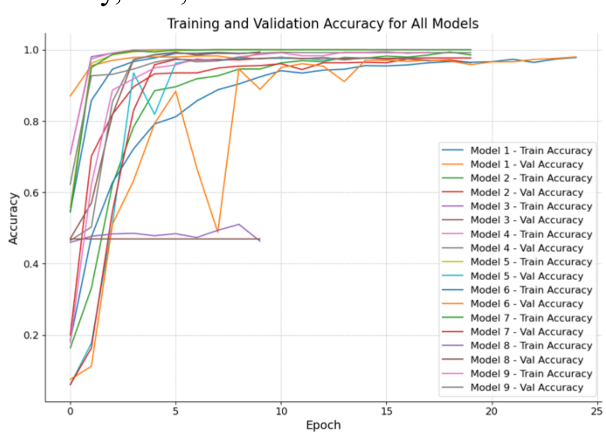
For the type classification task, we further enhanced the ResNet model (Model 6) by incorporating hyperparameter tuning and additional improvements. The key aspects include:

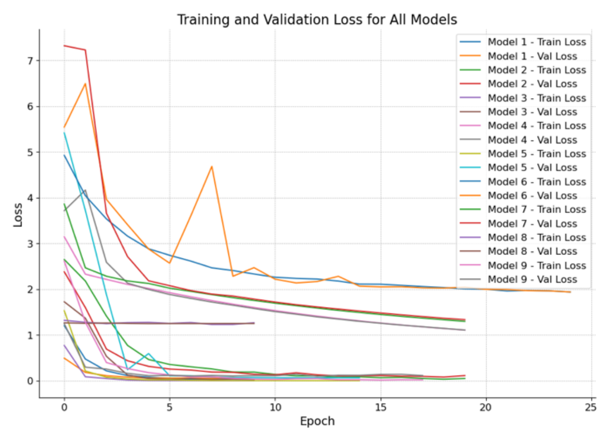
1. Data Augmentation
    a. Data augmentation is a technique used to increase the diversity and quantity of the training data by applying random transformations to the images.

b. We incorporated data augmentation techniques such as rotation, width and height shifts, shear, zoom, and horizontal flipping to generate augmented training samples.
c. Data augmentation helps the model to learn more robust features and reduces overfitting by exposing it to a wider range of variations in the training data.

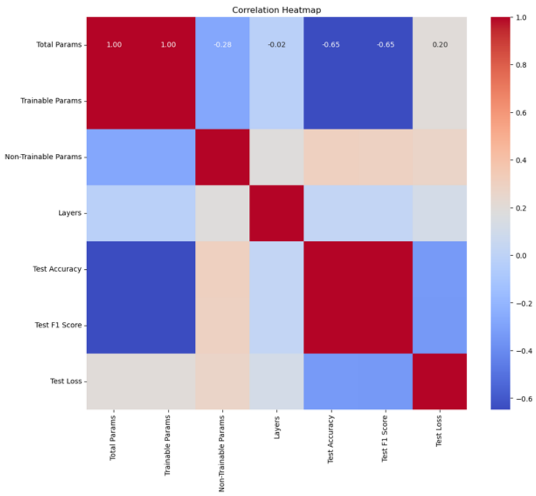
2. Dropout Regularization
   a. Dropout is a regularization technique that randomly drops out a fraction of the neurons during training, preventing the model from relying too heavily on specific features.
   b. We added a dropout layer with a dropout rate of 0.5 after the dense layer in the ResNet model.
   c. Dropout helps in reducing overfitting and improving the model's generalization ability by forcing it to learn more robust and redundant representations.

The impact of hyperparameter tuning and improvements on the type classification model's performance is significant as you can see from the following three plots that display the accuracy, loss, and f1 score across all of the models.





Correlation Analysis:



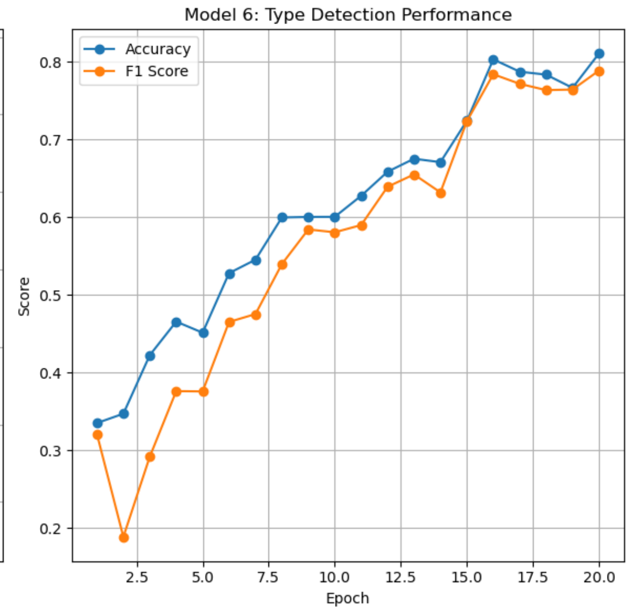
The correlation analysis revealed interesting insights into the relationship between model architecture and performance. The negative correlation between trainable parameters and test accuracy/F1 score suggests that simply increasing the number of parameters does not necessarily lead to better performance. Instead, it highlights the importance of careful model design and regularization techniques to prevent overfitting. The weak positive correlation between the number of layers and test accuracy/F1 score indicates that deeper networks alone may not guarantee improved performance. It emphasizes the significance of designing efficient architectures and leveraging techniques like residual connections to enable effective learning in deep networks. The analysis also showed a negative correlation between paramets per layer and test accuracy/F1 score, further reinforcing the notion that merely increasing model complexity does not always translate to better results. It underscores the need for a balanced approach, focusing on architecting models that can capture relevant features while maintaining generalization ability.

Independent Evaluation of Machine Learning Models:

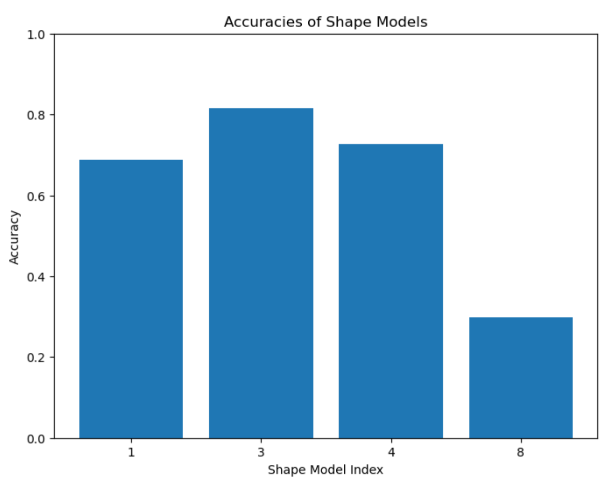
Dataset Creation

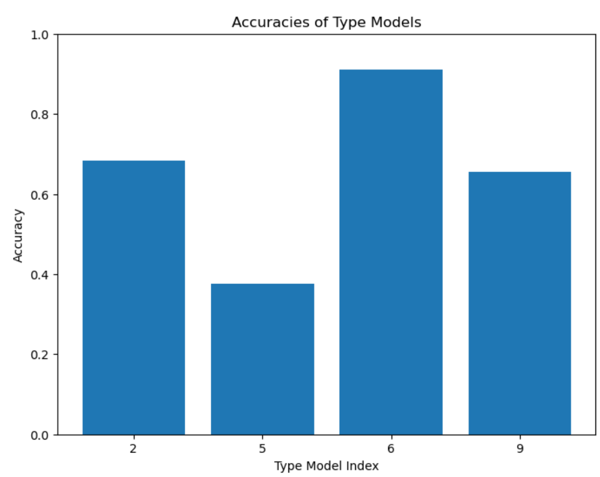
- We curated an independent dataset closely representing trafficsigns_dataset using German Traffic Sign Detection Benchmark dataset from Kaggle and supplementary images from Google search.
- Ensured comprehensive and diverse dataset for evaluating models' performance on each sign type.
- Challenging categories (noparking, laneend) required extra effort to find suitable images matching trafficsigns_dataset characteristics.

Model Performance Analysis:



- We chose best-performing shape and type models based on accuracy and F1 scores after considering hyperparameter tuning impact.
- Increasing the number of epochs as the images are being processed by our pre-trained model 3 and model 6 helped in improving the overall accuracy as shown in the above two graphs.





Best Shape Model: Shape Model 3 (Hyperparameter-Tuned ResNet Model)

- Top performer among shape models with 0.8164 overall accuracy
- Hyperparameter tuning (learning rate, L2 regularization, training epochs) enhanced performance
- Exceptional performance in classifying "hex" shape (0.9043 accuracy)

Success factors:

1. Enhanced learning capacity through optimized learning process
2. Improved regularization with L2 regularization to prevent overfitting and improve generalization
3. Increased training epochs for refined understanding of shape characteristics

Best Type Model: Type Model 6 (Hyperparameter-Tuned ResNet Model)

Key Points:

1. Clear winner among type models with 0.9098 peak validation accuracy and 0.9016 F1 score
2. Hyperparameter tuning process involved:
   - Optimizing learning rate for effective convergence and efficient training
   - Adjusting L2 regularization strength to prevent overfitting and improve generalization
   - Increasing training epochs to refine understanding of traffic sign types
3. Success factors:
   - Optimized ResNet architecture with deep residual connections for learning complex representations
   - Effective L2 regularization to prevent overfitting and improve generalization
   - Increased capacity to learn nuances of different traffic sign types through hyperparameter tuning
4. Demonstrated remarkable performance in traffic sign type detection
5. Hyperparameter tuning enhanced the model's ability to capture distinctive features and patterns
6. Results highlight the effectiveness of the tuning process for accurate and reliable traffic sign type classification

Conclusion:

In this project, we successfully designed, developed, and evaluated a traffic sign recognition system using various deep learning architectures. Through meticulous data preprocessing, model development, and hyperparameter tuning, we achieved promising results in both traffic sign shape detection and type classification tasks. The hyperparameter-tuned ResNet models emerged as the top performers, with Model 4 excelling in shape detection and Model 6 in type classification. The correlation analysis provided valuable insights, highlighting the importance of efficient model design and regularization techniques over simply increasing model complexity. The independent evaluation phase further validated the robustness and generalization ability of our best-performing models. By curating a diverse and representative dataset, we assessed the models' performance in real-world scenarios, with Model 3 demonstrating exceptional accuracy in shape detection and Model 6 showcasing remarkable performance in type classification. Overall, this project contributes to the advancement of autonomous driving technology by developing a reliable and accurate traffic sign recognition system. The insights gained from this study can be applied to other domains involving object recognition, emphasizing the significance of careful model design, hyperparameter tuning, and regularization techniques for optimal performance. Future work could explore the integration of this traffic sign recognition system into real-time autonomous driving systems, as well as investigating the potential of transfer learning and domain adaptation techniques to enhance the system's adaptability to different geographical regions and traffic sign variations.

Note: The main two notebooks to view are:

- Traffic_Model_Development.ipynb
- Indep_evaluation.ipynb

References:

a. Li, X., Zhao, Q., Tian, Y., & Wang, Y. (2018). Traffic Sign Detection and Recognition Using Fully Convolutional Network Guided Proposals. IEEE Transactions on Intelligent Transportation Systems, 19(6), 2042-2051.
b. Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., & Hu, S. (2020). Traffic-Sign Detection and Classification in the Wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2110-2118.