COMP30027 Report
Project 2: German Traffic Sign Prediction

1. Introduction

This project classifies German traffic signs into 43 distinct classes using four distinct machine learning models.

Data exploration was performed to examine the provided features, identifying necessary data cleaning and preprocessing steps to reduce dimensionality and extract additional engineered features. Two feature sets: the original provided features (120-dimensional) and a refined, engineered set (51-dimensional) were compared. The evaluation involved six classical baseline classifiers to select the more effective feature set.

Based on comparisons, the engineered features set proceeded for further training. I then fine-tuned Random Forest and Multi-Layer Perceptron (MLP) models through hyperparameter optimization and combined these optimized models with an Support Vector Machine (SVM) into a stacking ensemble, applying targeted class-aware adjustments to improve performance.

Finally, a Convolutional Neural Network (CNN) was developed, progressively enhancing its architecture and data augmentation techniques to achieve superior classification performance.

2. Methodology

2.1 Data exploration

Initial data exploration was exploited to understand the characteristics of the provided dataset. Metadata and associated features were loaded into tables, examining the class distribution and identifying significant class imbalances.

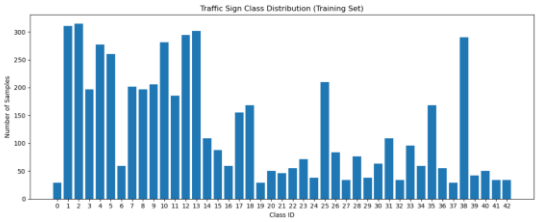

Figure 1- Class labels distribution.

Na-value checks indicated no missing values or zero-variance for any feature, eliminating the need for imputation. Feature distribution and correlation analysis were conducted, identifying skewness in the edge density feature.

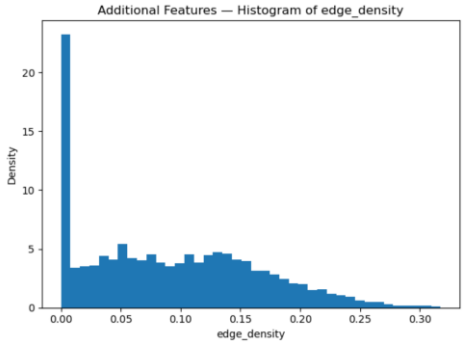

Figure 2- Edge density distribution.

Additionally, mean RGB values were converted to chroma RGB to focus on colour ratios rather than brightness.

2.2 Feature engineer

Understanding the current data state, feature engineering was implemented. First, local binary patterns (LBP) and Hu invariant moments were extracted for capturing local texture details (10 dimensions) and global shape descriptors (7 dimensions). The provided 96-dimensional color histograms were reduced to 10 PCA-based dimensions. Combined with pre-processed original features, the engineer features resulted in a computationally efficient 51-dimensional feature set (20-dimensional HOG PCA, 10-dimensional color histogram PCA, 3-dimensional chroma RGB, log-transformed edge density, 10-dimensional LBP, and 7-dimensional Hu moments).

2.3 Feature sets comparison

The provided raw feature set (120-dimensional) and the engineered feature set (51-dimensional) were compared by evaluating their performance on six classical baseline models: Logistic Regression, Random Forest, K-Nearest Neighbours (KNN), Support Vector Machine (SVM), Gaussian Naive Bayes, and Multi-Layer Perceptron (MLP), using a stratified 80/20 train-validation split. Results favoured the engineered feature set, which was then selected for further analysis.

2.4 Random Forest and Multi-Layer Perceptron (MLP) base models

Random Forest and Multi-Layer Perceptron (MLP) were selected as base models. The two classifiers were optimized using exhaustive grid search combined with stratified cross-validation. For MLP, feature selection was applied using a search for the best subsets of features, whereas Random Forest inherently managed feature selection. After the search, best performing parameters for each model are utilized. Optimized Random Forest and MLP classifiers significantly improved accuracy and macro F1 scores on the validation set.

2.5 Stacking (Ensemble model)

These optimized classifiers were then incorporated into a stacking ensemble (Wolpert, D.H., 1992), combined with an additional SVM classifier. Stratified 5-fold cross validation was implemented during training to reduce bias and the risk of overfitting.

Error analysis of the stacking model was implemented using classification report and heat map confusion matrix. It identified specific classes with recall below 80%, prompting targeted re-weighting strategies. The first tuning approach using reweighting low-recall classes showed noticeable improvement on validation set. Further refinements - including wrapping the model with a calibrated classifier, tuning the meta-classifier via grid search, and targeted oversampling - were explored but yielded minimal additional gains.

2.6 Convolutional Neural Network (CNN)

Finally, CNN was developed using raw image data (Krizhevsky, A., Sutskever, I., & Hinton, G.E., 2012). A small CNN model was trained and analysed. Learning curves identified sign of underfitting. Improvements included enhanced data augmentation techniques, increased model depth, expanded training epochs, and larger image resolutions. The CNN employed the Rectified Linear Unit (ReLU) activation function to introduce non-linearity, and gradient descent was used to iteratively optimize model performance. An adaptive learning rate was applied, which automatically reduced the learning rate when minimal performance improvements were observed. Additionally, appropriate early stopping criteria were implemented to prevent overfitting. These adjustments substantially improved performance, ultimately achieving the highest validation and test set accuracies among all explored models. Further tuning methods, such as increasing model capacity and regularization with a cosine-decay schedule, were also considered; however, these additional refinements did not yield improvements.

3. Results

3.1 Baseline models

Provided 120-dimension Feature Set		
Model	Accuracy	Macro F1
Logistic Regression	0.814208	0.800439
Random Forest	0.775046	0.759802
K-NN (k=5)	0.343352	0.297461
SVM	0.040073	0.024042
Gaussian NB	0.225865	0.236243
MLP (100 hidden units)	0.818761	0.794261

Table 1- Performance of baseline classifiers on the original provided 120-dimensional feature set.

Engineered 51-dimension Feature Set		
Model	Accuracy	Macro F1
Logistic Regression	0.781421	0.766727
Random Forest	0.780510	0.748588
K-NN (k=5)	0.537341	0.464084
SVM	0.200364	0.139245
Gaussian NB	0.581967	0.575645
MLP (100 hidden units)	0.775046	0.755813

Table 2- Performance of baseline classifiers on the engineered 51-dimensional feature set.

The engineered 51-dimensional set generally demonstrates improved stability and better performance.

3.2 Random Forest

Max depth	None
Max features	sqrt
Min samples split	5
Number of estimators	500

Table 3- Optimized hyperparameters for Random Forest obtained via grid search.

Mean Accuracy	0.8169
Mean Macro F1	0.8066

Table 4- Performance of optimized Random Forest using 5-fold stratified cross-validation on the training set.

Validation Accuracy	0.8115
Validation Macro F1	0.7820

Table 5- Performance of optimized Random Forest evaluated on the held-out validation set.

Table 5 show strong performance on the unseen validation set, confirming the optimized model generalizes effectively.

3.3 Multi-Layer Perceptron (MLP)

Select Features	30
Hidden layer size	(100, 50)
alpha	0.01
Learning rate	0.001
Activation	relu

Table 6- Optimized hyperparameters for MLP obtained via grid search.

Mean Accuracy	0.8400
Mean Macro F1	0.8265

Table 7- Optimized MLP evaluation on 5-fold stratified cross-validation on training set

Validation Accuracy	0.8333
Validation Macro F1	0.8033

Table 8- Optimized MLP evaluation on hold-out validation set

The validation results in Table 8 remain consistently strong, confirming that the optimized MLP model generalizes effectively to unseen data.

3.4 Stacking (Ensemble model)

3.4.1 Initial Model

Mean Accuracy	0.8899
Mean Macro F1	0.8860

Table 9- Initial Stacking evaluation on 5-fold stratified cross-validation on training set

Validation Accuracy	0.8789
Validation Macro F1	0.8728

Table 10- Initial Stacking evaluation on hold-out validation set

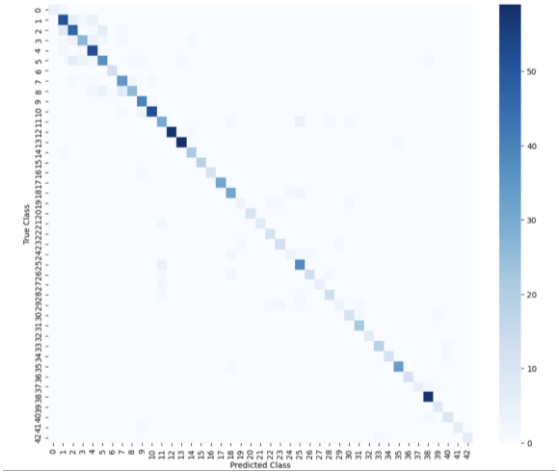

Figure 3- Stacking classifier confusion matrix

Class	Recall
29	38%
24	43%
19	50%
0	67%
8	67%
3	69%
5	71%
27	71%
2	75%
26	76%
21	78%

Table 11- Low performing classes (recall < 80%)

The ensemble achieves better performance compared to individual baseline models, showed by high accuracy and macro F1 scores.

3.4.2 Class-reweighting

Mean Accuracy	0.8807
Mean Macro F1	0.8733

Table 12- Stacking evaluation on hold-out validation set after reweighting low-recall classes

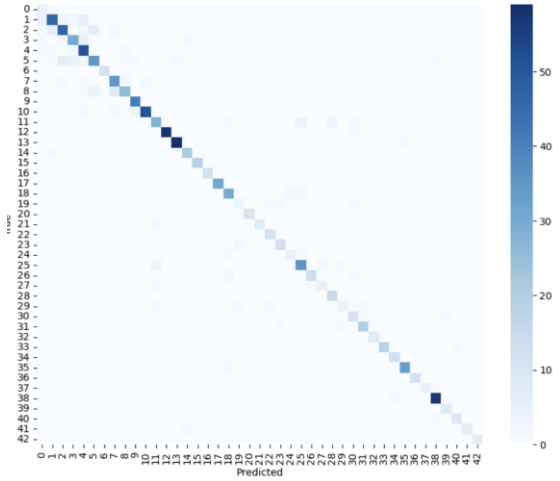

Figure 4- Stacking classifier confusion matrix after reweighting low-recall classes

The stacking model's performance slightly improved after applying targeted class reweighting (Table 11).

3.4.3 Further fine-tunings

Validation Accuracy	0.8652
Validation Macro F1	0.8567

Table 13- Stacking model evaluation with calibrated probability outputs on validation set

Estimator C	10
Estimator penalty	12

Table 14- Optimized hyperparameters for Stacking obtained via grid search.

Validation Accuracy	0.8770
Validation Macro F1	0.8714

Table 15- Optimized Stacking evaluation on hold-out validation set using grid search.

Validation Accuracy	0.8761
Validation Macro F1	0.8704

Table 16- Stacking evaluation on hold-out validation set after further targeted oversampling of low-recall classes

Further fine-tuning strategies did not surpass the performance of the earlier class-reweighting approach.

3.5 Convolutional Neural Network (CNN)

3.5.1 Small CNN

Accuracy	0.7220

Table 17- Small CNN evaluation on hold-out validation set

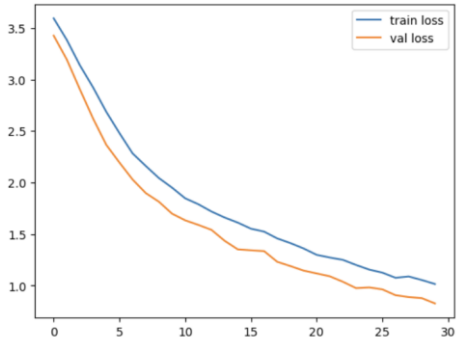

Figure 5- Small CNN loss curves

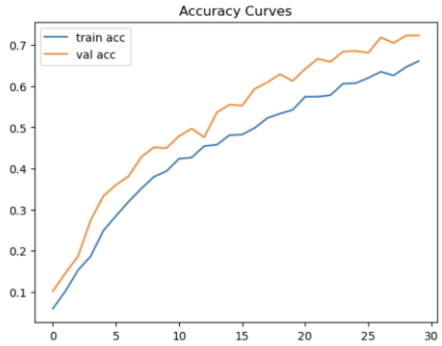

Figure 6- Small CNN accuracy curves

The small CNN model demonstrates signs of underfitting, indicating a need for further optimization.

3.5.2 Tuned CNN

Accuracy	0.9772

Table 18- Tuned CNN (augmentation techniques, increased model depth, expanded training epochs) evaluation on hold-out validation set

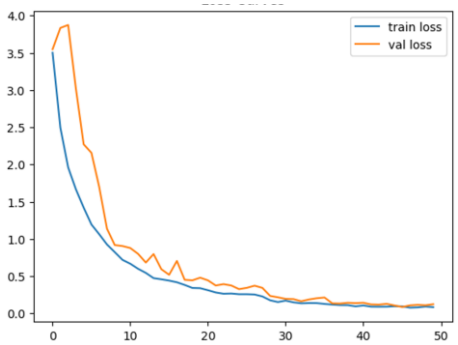

Figure 7- Tuned CNN loss curves

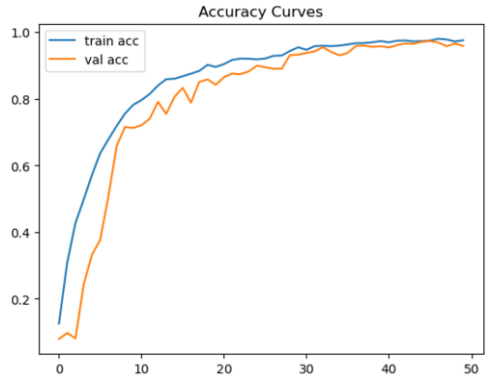

Figure 8- Tuned CNN accuracy curves

Accuracy	0.9754

Table 19- Further increased capacity CNN evaluation on hold-out validation set

Accuracy	0.9626

Table 20- Regularized CNN with cosine- decay schedule evaluation on hold-out validation set

The first tuned CNN model achieved the highest validation accuracy. Subsequent fine-tuning attempts did not lead to further performance improvements.

3.5.3 Final CNN

Accuracy	0.9863

Table 21- Best CNN, with upscaling image to 96 x 96, evaluation on hold-out validation set

The finalised CNN, implemented from the first tuned CNN model, with additional upscaling image, is the best performing model.

4. Discussion and Critical Analysis

4.1 Data exploration

During data exploration, class imbalances were observed: certain traffic-sign categories had over 300 training examples, while others were underrepresented with fewer than 30 instances. Such imbalances can bias classifiers toward majority classes. To counter this, stratified splits for both hold-out validation and cross-validation folds were adopted, ensuring that each fold retained representative samples of rare classes. Oversampling techniques and class-weighted loss functions were applied during model training, and tracked per-class macro F_1 to guard against ignoring low-frequency categories.

The analysis also identified strong skew in the edge-density feature. By applying a log transform, variance was stabilized and distribution was brought closer to symmetry, improving numerical conditioning. Similarly, mean RGB values were converted into chroma features to emphasize relative color information over absolute intensity.

4.2 Features engineering

From raw images, local binay patterns were extracted to capture fine-grained texture, and Hu invariant moments encoded global shape descriptors. The 96-bin color histograms were reduced to 10 principal components (chroma PCA) to remove redundancy.

This dimensionality reduction offers several advantages. First is compactness, which fewer inputs reduce model complexity and training time. Interpretability is another factor, where each feature group has a clear semantic meaning (shape, texture, color). This approaches also reduced overfitting risk - by removing collinear and noisy dimensions. Lastly, it reduced computational cost both in classical algorithms and grid-search.

4.3 Feature Sets Comparison

120-dimensional and 51-dimensional feature sets were compared based on six baseline classifiers (Table 1 and Table 2). Without normalization, SVM collapses (4% and 20% accuracy) due to scale sensitivity and the curse of dimensionality. This highlights the need for feature scaling in later pipelines. Non distance-based models (Logistic Regression, MLP) perform well on full-dimensional inputs, while distance-based methods (k-NN, SVM) and probabilistic models that assume feature independence (Gaussian Naive Bayes) benefit substantially from dimensionality reduction and decorrelation. Random Forest remains effective in both settings, reflecting its internal feature-selection and insensitivity to redundant inputs.

4.4 Random Forest Optimization

Random Forest was selected for its robustness against irrelevant features and strong performance in handling non-linear relationships and interactions among features. Feature selection was not explicitly included in the grid search because the algorithm performs embedded feature selection via its tree based split. The tuned model from grid search approach exhibits higher cross-validation accuracy and macro F1 (Table 4) and improved validation performance (Table 5) compared to the default model. This gain arises from better control over tree complexity, preventing both under and overfitting, which enhances generalization.

4.5 MLP Optimization

The MLP was chosen for its capability to model complex, non-linear patterns and adapt flexibly to diverse feature spaces. The model underwent feature selection followed by exhaustive grid search over layer sizes, regularization strength (alpha), and learning rate (Table 6). Its optimized configuration achieved strong cross-validation results (Table 7) and delivered consistent performance on the held-out validation set (Table 8). The use of ReLU activations and early stopping helped the network converge efficiently without overfitting, while scaling inputs ensured stable gradient descent updates across dimensions.

4.6 Stacking (Ensemble model) Optimization

Building on the optimized Random Forest and MLP, these two models were combined with an SVM in a stacking ensemble. The initial stacking model (Table 9 - 10) outperformed each individual base learner, validating the theory that stacking results are generally better than the best of the base classifiers.

Error analysis revealed classes with lower recall, prompting class-reweighting which yielded a modest lift in performance (Table 12). Further adjustments, including probability calibration, meta-learner hyperparameter tuning, and targeted oversampling, failed to surpass the class-reweighted configuration. In particular, probability calibration may have disrupted the ensemble's decision margins; hyperparameter tuning of the meta-learner offered limited flexibility once base predictions were optimized.

The confusion matrix for optimized stacking model highlights persistent confusions among similar speed-limit signs (e.g., 50, 60, 80km/h), suggesting that resolving fine-grained visual distinctions may require more detailed feature representations or an deep learning approach such as CNN-based models.

4.7 CNN Optimization

The small baseline CNN (Figures 5-6) exhibited underfitting, as evidenced by a persistent gap between training and validation curves and low validation accuracy (72.2%). The model's architecture was tuned with data augmentation, deeper convolutional layers, and extended training epochs, along with an adaptive learning rate schedule (ReduceLROnPlateau). This first tuned CNN achieved a substantial performance jump to 97.7% (Table 18), demonstrating a well-balanced fit where training and validation accuracies converge and stabilize (Figure 7 and Figure 8).

Subsequent modifications - increasing network capacity and applying cosine-decay regularization - did not yield further gains, confirming that the first tuned configuration was already near the optimal bias–variance trade-off.

Finally, the input resolution was upscaled from 32x32 to 96x96 and reached 98.6% validation accuracy. Retrained on the full dataset, it achieved 99.3% accuracy on the hidden test set submitted to Kaggle. The remaining errors predominantly occur on speed-limit signs and extremely blurred images, which pose inherent ambiguity even for human annotators.

5. Conclusion

The project has shown that careful data exploration and preprocessing can substantially improve the performance of classical classifiers. Grid search and cross-validation refined Random Forest and MLP models, and a stacking ensemble further refined their

complementary strengths. In parallel, a tuned CNN achieved the highest accuracy, showing the power of deep learning at the expense of greater computational cost.

The comparison illustrates a key trade-off: classical models with engineered features provide interpretable decisions and efficient training, while CNNs require more resources but deliver near-human performance. Remaining misclassifications, often involving similar speed-limit signs or heavily blurred images, implementing the natural limits of the data.

While model architecture and training strategy were effectively optimized, limitations still existed. Absolute perfection is constrained by data quality and intrinsic visual overlap among classes. Future enhancements could involve obtaining higher-quality, more diverse image datasets or applying advanced data augmentation strategies. Recognizing these practical constraints helps guide realistic expectations and strategic decisions for further improvement.

6. References

Wolpert, D.H. 1992. Stacked generalization. *Neural Networks*, 5(2), 241-259.

Krizhevsky, A., Sutskever, I., & Hinton, G.E. 2012. ImageNet classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 25, 1097-1105.