COMP30027 Project2 Report
Traffic Sign Prediction

1 Introduction

Traffic sign prediction is essential for intelligent transportation systems and autonomous driving. This project compares traditional machine learning models and deep learning approaches for classifying traffic signs. After extracting and selecting diverse image features, models including K-Nearest Neighbors (KNN), Extreme Gradient Boosting (XGBoost), Support Vector Machine (SVM), Random Forest, and a Convolutional Neural Network (CNN) were trained. CNN achieved the highest test accuracy (99.30%), demonstrating the superior performance of deep learning in image-based classification.

2 Methodology

2.1 Dataset

A subset of the German Traffic Sign Recognition Benchmark (GTSRB) was used in this study. The dataset includes:

- 5,488 training images with class labels across 43 traffic sign categories
- 2,353 unlabeled test images

In addition to raw image data, the dataset provides several pre-extracted features:

- Histogram of Oriented Gradients (HOG), reduced using Principal Component Analysis (PCA)
- Color histograms
- Additional features such as edge density, texture variance, and mean RGB values

2.2 Feature Extraction

To enhance classification performance beyond what the original dataset provided, additional shape and texture features were extracted to enrich image representation.

Shape features were computed based on the largest contour in each image, including geometric descriptors such as circularity, aspect ratio, convexity, solidity, extent, and normalized central moments (up to the third order) for scale- and rotation-invariant representation.

Texture features included Local Binary Patterns (LBP), from which entropy, energy, uniformity, and maximum bin values were derived. Gray-Level Co-occurrence Matrix (GLCM) features – contrast, homogeneity, correlation, and energy – were calculated in both horizontal and vertical directions. Additionally, gradient-based features such as the mean and standard deviation of gradient magnitudes and directional histogram statistics were included.

2.3 Feature Processing

The extracted features, combined with those originally provided, covered a broad range of dimensions including shape, texture, color, edge, gradient, and HOG. To reduce dimensionality, eliminate redundancy, and improve model performance and interpretability, feature selection was applied. This step aimed to identify the most discriminative features for traffic sign classification.

2.3.1 Data Splitting

The training data was split into training and validation sets using an 80:20 ratio with stratified sampling to preserve class distribution across subsets.

2.3.2 Feature Scaling

All features were standardized using z-score normalization (zero mean, unit variance) to ensure consistency and support effective model training. This step is essential for models sensitive to feature scale, such as SVMs and Lasso regression.

Although Random Forest (RF) is scale-invariant and Mutual Information (MI)-based selection does not strictly require standardiza-

tion, applying it uniformly ensures consistency across all feature selection methods and supports fair performance comparisons.

2.3.3 Feature Selection

Three methods were employed to select the most informative features, and their effectiveness was later evaluated through model performance.

A **RF** classifier with 100 trees was trained on the standardized feature set. Feature importance was estimated based on the average impurity reduction each feature contributed across the ensemble (Breiman, 2001), and those above the mean value were retained. This process selected 32 features (Figure 1), with the top-ranked features primarily coming from the PCA-reduced HOG descriptors.

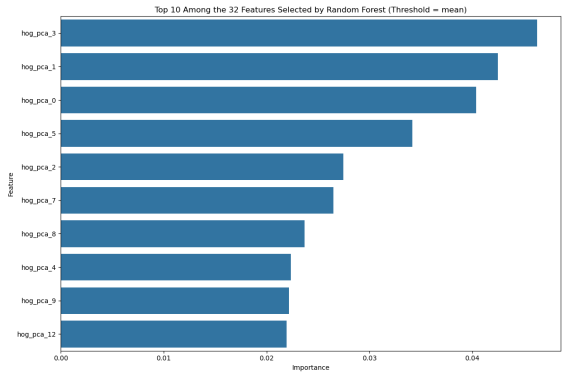

Figure 1: Top 10 Features Ranked by Importance Among the 32 Selected by RF

Lasso is a linear model that performs feature selection by applying an L1 penalty, which drives less important feature coefficients to zero (Ng, 2004). In this study, a Lasso model with a regularization strength of $\alpha = 0.01$ was applied to the standardized features, resulting in 129 non-zero coefficients (Figure 2). Notably, global color features, particularly from the blue and red channels, emerged as important predictors.

MI was used as a filter-based method to assess the dependency between each feature and the class label, capturing both linear and nonlinear relationships. As a model-agnostic approach, MI provides a general measure of feature relevance for classification (Peng et al., 2005). In this study, the top 30 features with the highest MI scores were selected (Figure 3), with the highest-ranking features primarily from PCA-reduced HOG descriptors, followed by global color features from the blue and

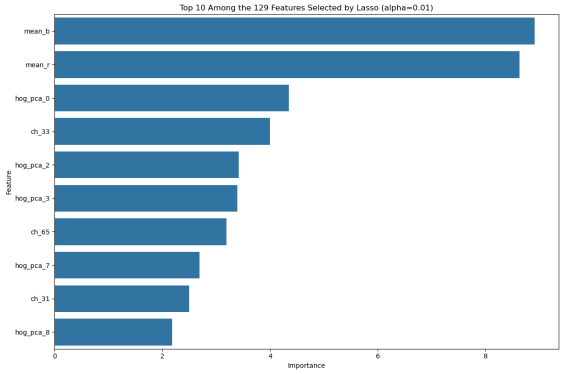

Figure 2: Top 10 Features Ranked by Importance Among the 129 Selected by Lasso Regression

red channels.

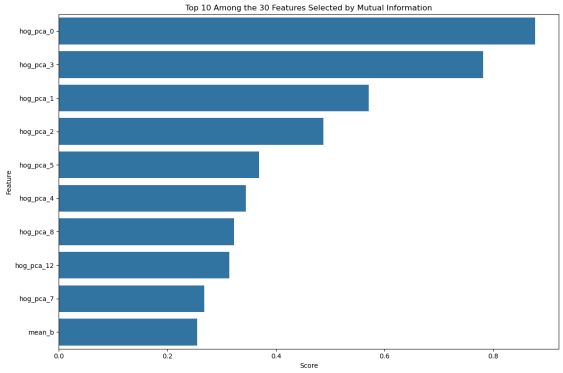

Figure 3: Top 10 Features Ranked by Importance Among the 30 Selected by MI

Across all three methods, the top-ranked features were consistently PCA-reduced HOG descriptors and global color features, particularly from the red and blue channels. This is likely because traffic signs possess distinct edge and gradient structures, which are effectively captured by HOG features. In addition, color plays a key role in sign classification: red is commonly used for warning and prohibitory signs, while blue typically indicates mandatory actions.

2.4 Model Training

For each feature selection method, the selected features were standardized using z-score normalization before training. All models were trained and evaluated on the validation set using accuracy as the performance metric. The classifiers used in this study are outlined below:

2.4.1 KNN

KNN is a non-parametric, instance-based learning method that classifies a sample based on the

majority class among its k nearest neighbors in the feature space (Cover and Hart, 1967). Hyperparameters used:

- Number of neighbors $k = 5$
- Distance metric: Euclidean

2.4.2 XGBoost Classifier

XGBoost is a scalable, tree-based ensemble algorithm using gradient boosting (Chen and Guestrin, 2016), known for its robustness and high performance in classification tasks. Hyperparameters used:

- Number of estimators = 100
- Learning rate = 0.1

2.4.3 SVM

SVM seeks to find the optimal hyperplane that maximizes the margin between different classes. It performs well in high-dimensional spaces and is effective with non-linear decision boundaries (Boser et al., 1992). Hyperparameters used:

- Kernel = Radial Basis Function (RBF)
- Regularisation parameter $C = 1.0$
- Kernel coefficient $\gamma = scale$
- Probability estimates enabled

2.4.4 RF Classifier

RF is an ensemble of decision trees that aggregates predictions via majority voting (Breiman, 2001). It also provides internal estimates of feature importance. Hyperparameters used:

- Number of trees = 100
- Criterion = *gini*
- No maximum depth, nodes are expanded until pure
- Max features = *sqrt*
- Bootstrap samples used when building trees
- Parallel processing enabled with $n\_jobs = -1$

To further refine model performance, hyperparameter tuning was subsequently conducted on the most promising feature–model pair.

2.4.5 CNN

To address the performance limitations of traditional machine learning models, a baseline CNN was implemented to learn directly from raw image data. CNNs are well-suited for visual recognition tasks due to their ability to capture spatial hierarchies through convolutional layers (Lecun et al., 1998).

Images were resized to 48×48 pixels with three RGB channels. To enhance generalization, training images were augmented via random rotations ($\pm 10°$), affine transformations, and color jitter. All images were normalized to have a mean and standard deviation of 0.5 per channel, scaling pixel values to $[-1, 1]$.

The model consisted of three convolutional blocks (Conv $\rightarrow$ ReLU $\rightarrow$ BatchNorm $\rightarrow$ MaxPool), followed by a fully connected layer with dropout and a final softmax output for classifying 43 traffic sign categories. Training used the Adam optimizer with early stopping based on validation accuracy. Hyperparameters used:

- Batch size = 64
- Learning rate = 0.001
- Max epochs = 50
- Dropout rate = 0.5
- Early stopping patience = 10 epochs

A *ReduceLROnPlateau* scheduler decreased the learning rate by a factor of 0.2 when validation loss plateaued for 5 epochs. The best-performing model on the validation set was saved for testing.

3 Results

Following the validation results (see Table 1), the SVM classifier trained on features selected by RF was identified as the best-performing model (the confusion matrix is shown in Figure 4). To further optimize its performance, a hyperparameter tuning process was conducted using a 5-fold cross-validated Randomized Search on the combined training and validation sets.

The tuning explored different values for the regularization parameter C, kernel type, and kernel coefficient γ, with the following search space:

- $C : [0.1, 1, 10, 100]$
- $\gamma : [scale, auto, 0.01, 0.1]$

	RF	Lasso	MI
KNN	74.68	62.20	67.85
XGBoost	78.42	79.51	76.59
SVM	81.15	76.68	75.23
RF	80.05	80.15	78.32

Table 1: Validation Accuracy (%) of Classifiers by Feature Selection Method
(Row Labels: Classifiers; Column Labels: Feature Selection Methods)

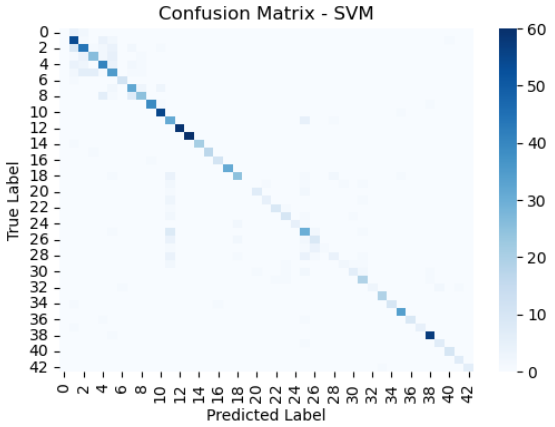

Figure 4: The Confusion matrix of SVM trained with RF-selected features

- Kernel: $[rbf, poly, sigmoid]$

The best configuration identified was: $\{\text{Kernel} = RBF, \gamma = scale, C = 10\}$.

This tuned model was then applied to the test set, and the predictions were submitted to Kaggle for evaluation. Despite promising validation results, the model achieved a test accuracy of only 33.93%, indicating poor generalization. This highlights the limitations of traditional machine learning methods in handling complex visual patterns, motivating the exploration of deep learning approaches.

To address this, a CNN model was implemented, demonstrating a much stronger learning capacity (Figure 5 and 6). During training, accuracy rose sharply within the first five epochs, accompanied by a rapid decline in loss. From epoch 6 to 37, performance continued to improve more gradually, reaching a peak validation accuracy of 99.36% at epoch 37. After that, the model's performance plateaued, and training was stopped at epoch 47 through early stopping based on validation performance, effectively preventing overfitting. When evaluated on the test set, the final CNN model achieved an impressive accuracy of 99.30%, significantly outperforming traditional methods and demonstrating strong generalization as well as robustness in extracting and interpreting complex image features.

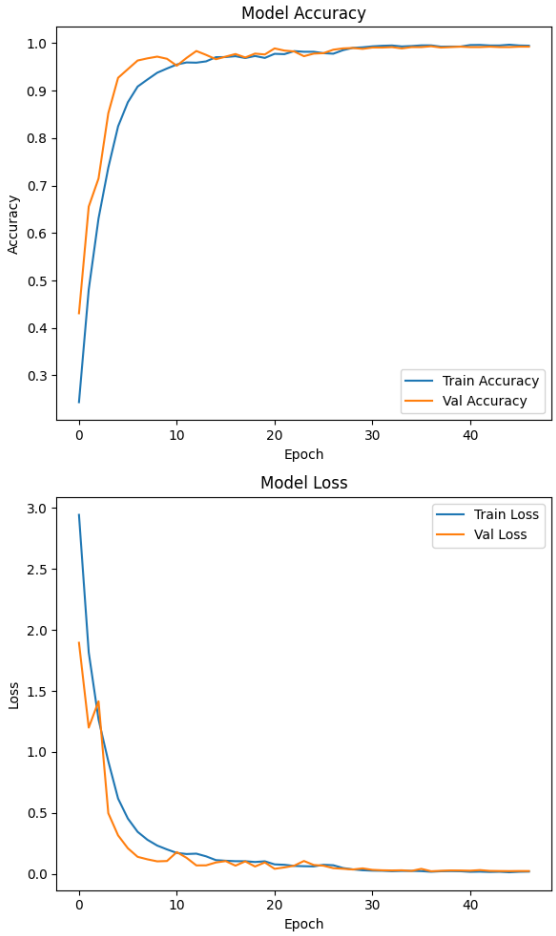

Figure 5: Training and validation accuracy and loss curves for the CNN model

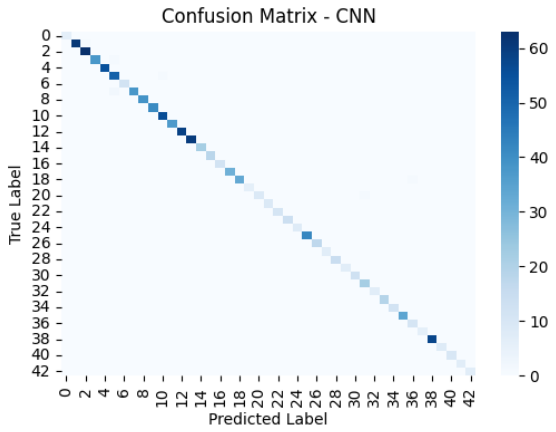

Figure 6: The Confusion Matrix of the best CNN model on Validation Set

4 Discussion and Critical Analysis

In this project, three feature selection methods – RF, Lasso, and MI – were employed to reduce dimensionality and enhance model interpretability. Among them, lasso retained the largest number of features by preserving correlated predictors, while MI and RF were more selective, prioritizing features with strong individual predictive power. Notably, the best validation performance was achieved by an SVM trained on RF-selected features, despite the subset being more compact. This suggests that effective feature selection is not solely about quantity but alignment with the learning algorithm's inductive bias. In the case of SVMs, which are sensitive to irrelevant or redundant features, RF likely provided a more focused, discriminative feature set that enhanced generalization.

The behavior of the tested models further reveals a fundamental contrast between traditional machine learning methods and deep learning in addressing image classification.

KNN, as a non-parametric, instance-based learner, depends on a distance metric (Euclidean distance in this case) that becomes unreliable in high-dimensional spaces due to the curse of dimensionality. Its lack of an explicit training phase and high sensitivity to noise contribute to instability and poor generalization.

XGBoost and RF, both ensemble methods based on decision trees, are robust against variance but inherently treat features independently. This assumption hinders their ability to model spatial locality or structural relationships between feature – an essential requirement in visual tasks. Even when rich handcrafted features such as shape or texture are provided, these models are unable to capture how such local cues compose into global patterns, which may result in either overfitting or underfitting depending on feature interactions and tree complexity.

SVM offers strong generalization guarantees grounded in margin theory, aiming to minimize structural risk by maximizing the decision margin between classes. However, linear SVMs cannot capture the complex, non-linear spatial patterns common in visual data. While nonlinear kernels like RBF improve expressiveness, they introduce significant computational cost, especially as data size and dimensionality increase. SVMs also scale poorly in multi-class settings, relying on one-vs-one or one-vs-rest schemes that add to model complexity. Crucially, when applied to high-dimensional handcrafted features, such as shape, edge, and texture descriptors, SVMs are prone to overfitting spurious patterns. This is evident in the gap between validation accuracy (81.15%) and test accuracy (33.93%), highlighting limited generalization and the lack of inductive bias toward visual structures.

In contrast, CNN learns hierarchical representations directly from raw pixels through stacked convolutional and pooling layers. This architecture exploits spatial locality and enables the network to detect increasingly abstract patterns, from edges to object parts and full shapes, without manual feature engineering. As a result, the CNN achieved a significantly higher test accuracy (99.30%), showcasing superior generalization and robustness.

From the confusion matrices (Figure 4 and 6), both the SVM and CNN models display a strong diagonal structure, suggesting that most predictions align with the true class labels. However, the SVM confusion matrix shows significantly more misclassifications than CNN, particularly in lower-index classes (e.g. class 11), despite these classes having more samples in the validation set. Intuitively, models tend to benefit from larger class sizes, but in this case, the opposite trend was observed. This can be attributed to the SVM's reliance on manually selected features, which may include redundant or noisy dimensions. When the number of samples in a class increases, the feature variation within that class also grows, making it harder for SVM to maintain a clean decision boundary. As a result, the model may overfit to subtle, non-generalizable patterns in the training data. In contrast, the CNN model, which learns hierarchical and spatially structured features directly from raw pixels, remained consistently accurate across all classes. Its ability to extract robust and abstract representations allowed it to generalize well even in the presence of intra-class variability.

5 Conclusion

This project addressed the problem of traffic sign recognition by comparing traditional machine learning approaches based on manually extracted features with a deep learning solution using CNN. Although traditional models such as SVM and RF achieved moderate performance with carefully selected features, their lack of

spatial inductive bias and reliance on static representations limited their generalization capability. In contrast, the CNN model, trained directly on raw image data, was able to automatically learn hierarchical and spatially meaningful features, achieving a significantly higher test accuracy of 99.30%. This big performance difference highlights the suitability of deep learning methods for visual recognition tasks, where the ability to extract and integrate local and global patterns is crucial. The results confirm that modern neural architectures, when applied correctly, offer a powerful framework to solve real-world vision problems such as traffic sign classification.

References

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45:5–32.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

T. Cover and P. Hart. 1967. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Andrew Y. Ng. 2004. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the Twenty-First International Conference on Machine Learning*.

Hanchuan Peng, Fuhui Long, and C. Ding. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238.