Machine Learning COMP30027 Assignment 2
Aditya Sarkar - 1354041

Introduction

This report explores supervised machine learning methods to classify German traffic signs from a subset of the GTSRB dataset. The task involved converting image data into numerical vector representations using various techniques, including canny edge detection, feature selection, and convolutional neural network embeddings. Subsequently, four classifiers, Random Forest, AdaBoost, SVM, and KNN, were evaluated through cross-validation and Kaggle testing metrics. The analysis investigates the effectiveness of each vectorisation approach, model accuracy, generalisability, and the impact of dimensionality reduction techniques. Overall, the objective was to critically examine and identify optimal methods for accurate and robust traffic sign classification.

Method

The pipeline for model training involved two main phases, first the images needed to be vectorised. In other words, they needed to be converted from a jpeg format where information is stored as pixels, to a vector, where information is stored as a list of floating point numbers each representing a different feature or aspect of the image. There are a variety of ways to vectorise an image which all involve various tradeoffs, but the ultimate goal is to compress the information within the image, keeping distinguishing information while removing unnecessary information. The provided feature set already included various features such as colour histograms, histograms of oriented gradients and edge density. Multiple different vectorisers were created to trial and evaluate model performance. Additional features related to edge placement in the images were extracted using canny edge detection. These additional features were combined with the provided featuresets to create three of the initial vectorisers as follows. “hybrid_vectoriser” which combines the features and then processes them using PCA dimensionality reduction to fit 500 features. “hybrid_vectoriser_np” is the same as “hybrid_vectoriser” but without the PCA. “hybrid_vectoriser_fs” again combines the features but then reduces the number of features to 500 by selecting the 500 most predictive features. Finally, a pre-trained convolutional neural network vectoriser, “img2vec_vectoriser” was also used to vectorise the images.

The second phase is the training of a machine learning model on the vectorised images, validating the model and then testing the model. Four models were tested including Random Forest, AdaBoost, SVM and KNN models. To validate the models before they were tested on the test dataset, 10 fold cross validation was performed on each vectoriser-model combination.

Results

Below in Figure 1 is the cross-validation results of each of the vectoriser-model combinations.

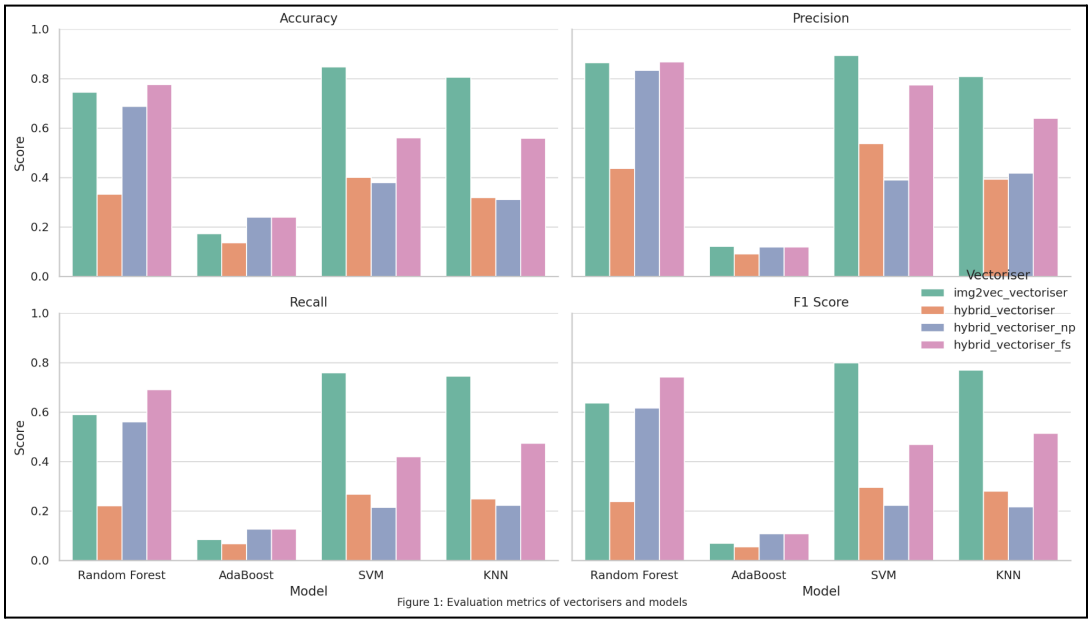

Figure 1: Evaluation metrics of vectorisers and models

Model	Accuracy	Precision	Recall	F1 Score	Average Score
Random Forest	63.56%	75.07%	51.58%	55.81%	61.50%
SVM	54.70%	64.89%	41.53%	44.69%	51.45%
KNN	49.85%	56.46%	42.29%	44.50%	48.27%
AdaBoost	19.67%	11.24%	10.10%	8.51%	12.38%

Table 1

Broadly, the random forest classifier performed the best out of the four models with an average score (across accuracy, precision, recall and F1 score) of 61.5% while the AdaBoost model performed particularly poorly with an average score of just 12.38% (See Table 1). The convolutional neural network (img2vec) vectoriser performed the best overall with an average score of 60.72% while hybrid_vectoriser_fs had an average score of 50.49% (see Table 2).

Vectoriser	Accuracy	Precision	Recall	F1 Score	Average Score
img2vec_vectoriser	64.31%	67.22%	54.50%	56.85%	60.72%
hybrid_vectoriser_fs	53.33%	59.99%	42.79%	45.83%	50.49%
hybrid_vectoriser_np	40.44%	44.00%	28.10%	29.11%	35.41%
hybrid_vectoriser	29.69%	36.44%	20.11%	21.71%	26.99%

Table 2

While the highest performing individual model and vectoriser combination was the image2vec vectoriser paired with an SVM, SVM performed substantially worse than random forest when using any of the hybrid_vectoriser models. Conversely, the RandomForest model was more robust, with the hybrid vectoriser with feature selection performing better than the img2vec vectoriser across the board.

Notably, PCA was not always useful to the performance of models. PCA was beneficial to both KNN and SVM models, with the models performing marginally worse with the Hybrid vectoriser with no PCA than with PCA. However, removing PCA substantially deteriorated the performance of both the RandomForest and AdaBoost models (likely because they both use decision trees). After removing PCA from the vectoriser pipeline the average performance score with the RandomForest model increased from 30.7% to 67.5% which is substantial.

Discussion

Out of the four models used, it would be expected that RandomForest and SVM perform the best as

Adaboost was by far the worst performer with its overall score being 12.38% compared to RandomForest's 61.5%. AdaBoost is an ensemble model which iteratively trains a number of weak decision trees with low depth and combines them to ultimately classify unseen instances. As a result of this ensemble method being built on many weak learners which inherently struggle with multi class classification, the model as a whole also struggles with multi class classification. Furthermore, in a multiclass problem, the error reweighting process critical to the AdaBoost model becomes unstable because reweighting based on misclassifications becomes less targeted thus challenging model generalisability. It can be observed below in Figure 2 that the Adaboost model predicted only three classes the majority of the time (5, 8 and 31) which is expected given the model is better suited to binary class problems as opposed to multi class problems.

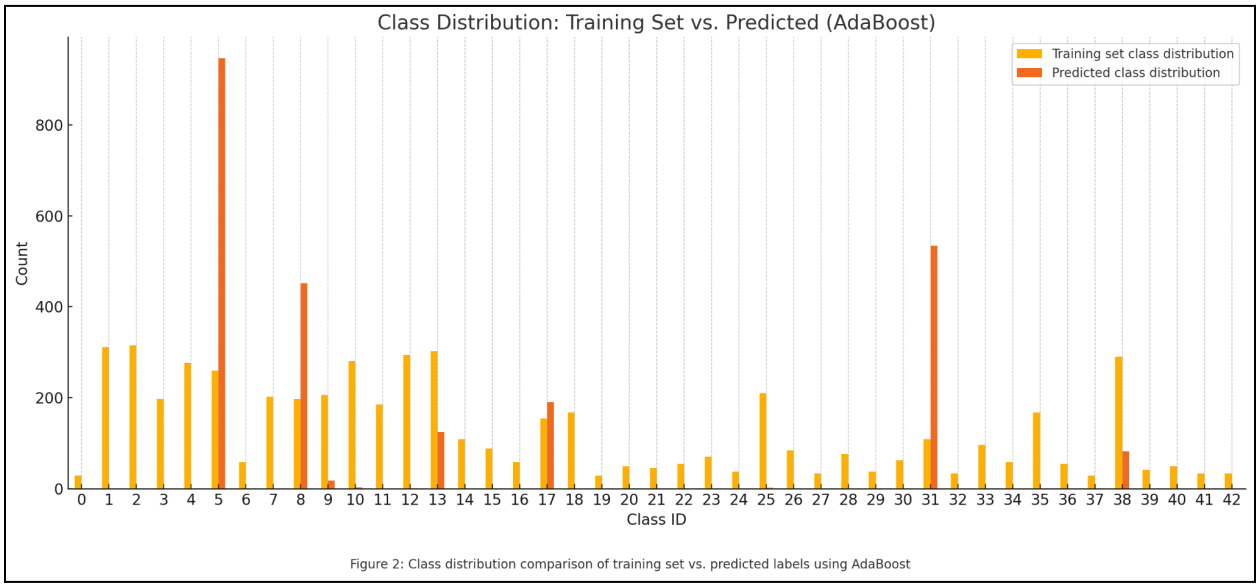

Figure 2: Class distribution comparison of training set vs. predicted labels using AdaBoost

Conversely, RandomForest (with the feature selecting hybrid vectoriser) (as well as the other examined models) have a predicted class distribution which more closely matches the training set. Upon closer inspection, however, it seems that the Random Forest model has a strong tendency to over-classify classes which are more frequently seen in the training set and under-classify classes which are rarer in the training set (See Figure 3 & 4).

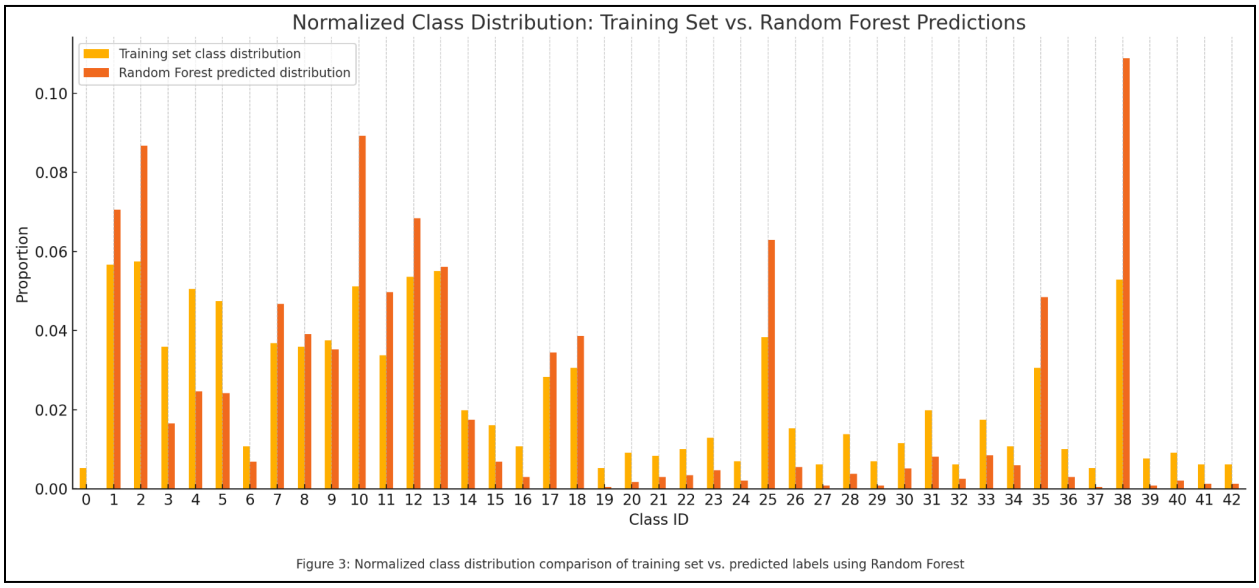

Figure 3: Normalized class distribution comparison of training set vs. predicted labels using Random Forest

To quantify this relationship a regression was performed between the prediction bias of classes and their proportion in the training set yielding a moderate correlation of 0.53. This implies that a lot of the accuracy gains of the RandomForest model originate from the models tendency to over predict frequent classes. While the model performed well in the cross validation, the model predicted frequent classes more accurately, boosting accuracy, but underperformed on rarer classes. If tested on a sample with a different class distribution, this behaviour would harm class level recall and F1 scores. The prediction biases suggest the model is less generalisable as it relies on the learned class distributions from the training set and indicates mild overfitting.
This is evident when examining the kaggle score for the RandomForest model as the score decreases to just 52.8%. When analysing the SVM model (with the feature selecting hybrid vectoriser) in the same way, the correlation between class frequency and prediction bias was significantly lower at just 0.17 (See Figure 4). This suggests that compared to the RandomForest model, the SVM model has more balanced predictions across classes and is less dependent on training frequency. Broadly, the SVM performs more equitably across rarer classes than the RandomForest model. Overall, the correlation results suggest that the SVM is a more generalisable model with less overfitting. This is confirmed by the SVM kaggle score of 53.2%, which is higher than the RandomForest kaggle score, despite the model performing significantly worse in cross validation.

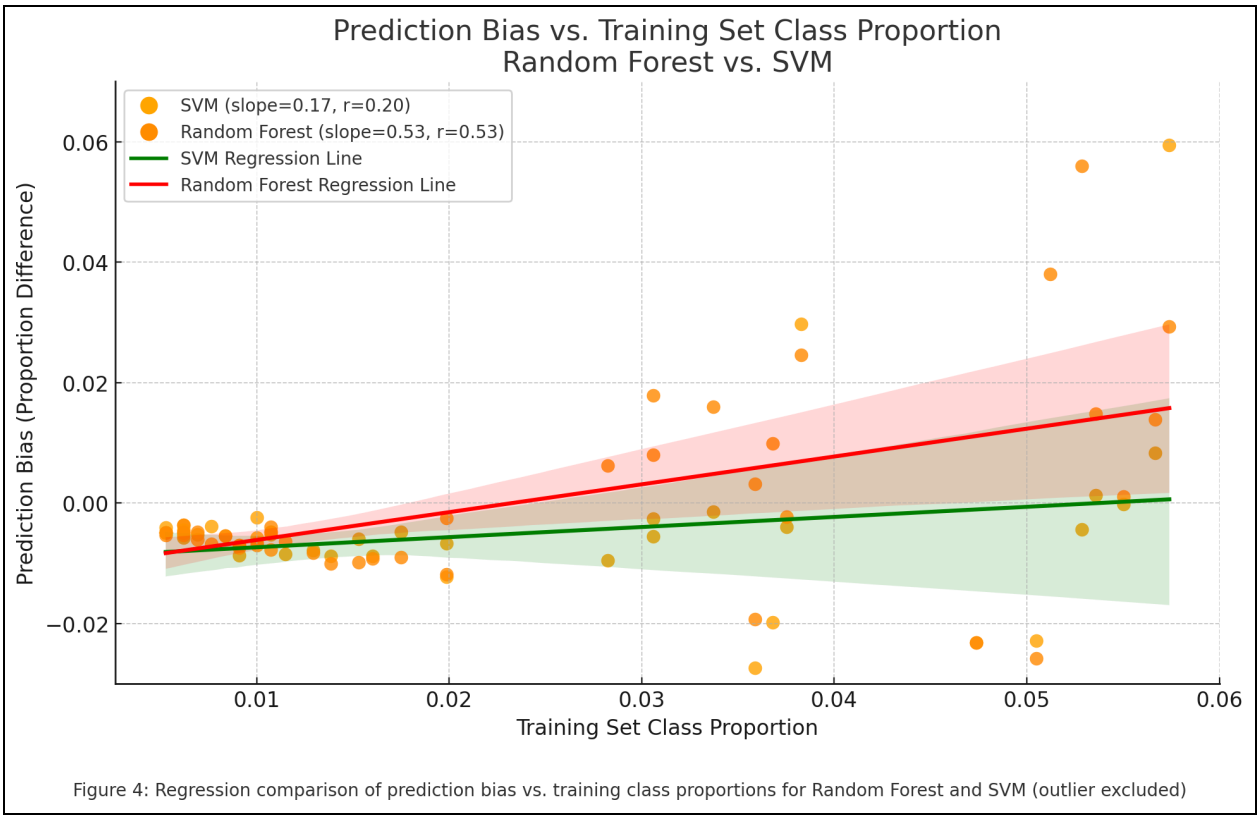

Figure 4: Regression comparison of prediction bias vs. training class proportions for Random Forest and SVM (outlier excluded)

Recall that “hybrid_vectoriser_np” did not use PCA while “hybrid_vectoriser” does. PCA significantly improved the performance of the SVM and KNN models by reducing dimensionality, eliminating noise, and capturing the directions of maximum variance in the data. For SVM, this helped produce cleaner margins by removing irrelevant or collinear features, enhancing its ability to find optimal separating hyperplanes. KNN, a distance-based algorithm, also benefited as PCA reduced the "curse of dimensionality" and made distance calculations more meaningful in the compressed feature space, leading to improved classification accuracy.

In contrast, PCA hindered the performance of Random Forest and AdaBoost. These tree-based models are inherently robust to high-dimensional and noisy data, as they perform automatic feature selection and model complex nonlinear interactions. PCA, by projecting original features into abstract linear combinations, removed useful interactions and interpretability that decision trees leverage for optimal splitting. As a result, the transformed feature space was less conducive to the hierarchical and threshold-based learning processes of tree ensembles, reducing their effectiveness.

The feature selection vectoriser “hybrid_vectoriser_fs” also significantly improved the performance of SVM and KNN models but only marginally improved the performance of the RandomForest model (See Figure 1). As mentioned previously, this reflects the native ability of the decision tree algorithm to choose the most useful features first, thus they are able to perform well even without feature selection during preprocessing. KNN and SVM models, on the other hand are more sensitive to noise and thus PCA benefits their performance significantly.

Conclusion

Through evaluation and error analysis, Random Forest emerged as the highest-performing model in cross-validation, though its performance declined during Kaggle testing due to bias toward frequent classes. Conversely, SVM showed greater consistency and generalisability despite lower cross-validation scores. PCA notably enhanced performance for SVM and KNN by reducing noise and dimensionality but was detrimental to tree-based models like Random Forest and AdaBoost, highlighting the importance of method-specific preprocessing. Ultimately, successful classification of traffic signs demands careful consideration of vectorisation techniques and model characteristics, underscoring that model robustness and generalisability are critical beyond initial accuracy scores.