COMP30027: Machine Learning
Assignment 2 Report

Thanh Dat Nguyen

1 Introduction

There is an increasing body of research conducted on improving the capabilities of Advanced Driver Assistance Systems (ADAS), and more recently, exploring the possibilities of self-driving vehicles. One very important aspect of these technologies is the ability to follow traffic rules, which includes recognising and identifying traffic signs through computer vision. Using the provided subset of the German Traffic Sign Recognition Benchmark (GTRSB) as the dataset, this report aims to explore this task by extracting shape-related features like Hu Moments and contours from the images, and using them alongside common classification Machine Learning techniques, specifically Support Vector Machines (SVM), Random Forests (RF), and Convolutional Neural Networks (CNN), to classify traffic signs.

2 Methodology

2.1 Feature Extraction

2.1.1 Contour Features

The first features considered were contours, which can be understood as curves that outline the boundary of an object, that can then assist in tasks like shape detection and analysis. In order to extract contours from the given dataset, all images were first resized to a width of 25, while still maintaining their aspect ratio to make the data more uniform while not necessarily altering their original shapes. After this, the images were grayscaled, blurred using Gaussian Blur, so that image thresholding using Otsu's method (Otsu, 1979) can be applied more effectively and with reduced noise. Afterwards, all contours were extracted, and the largest contour were chosen for further analysis. All of these steps were performed with the OpenCV library.

From the largest contour, the following features were extracted:

- **Aspect Ratio**: The contour's width divided by its height.
- **Extent**: The contour's area divided by its bounding box's area.
- **Solidity**: The contour's are divided by its convex hull's area.
- **Equivalent Diameter**: Diameter of a circle that has the same area as the contour's.
- **Orientation**: The angle of the contour's object.

2.1.2 Hu Moments

Image moments, in computer vision, capture information about objects in the image, such as its area or centroid, through the weighted averages of the image pixels' intensities. Similarly, Hu Moments attempt to capture identities about the shape of an object by calculating 7 moment invariants (Hu, 1962). This serves as a more concrete, mathematical alternative to contours, since they may be more prone to inaccuracies.

All seven Hu Moments were extracted using the OpenCV library, and were used as seven separate features.

2.2 Preprocessing

For images where certain features could not be extracted or calculated, its feature values were considered 0. Both provided and extracted features were then joined into one csv file, for a total of 132 features.

2.3 Feature Selection

Feature selection metrics chosen were ANOVA F-value and Mutual Information between features and label, provided by the sklearn library. The Chi Square metric was also considered, but rejected since it only applied to positive feature values. Since these are filtering methods and do not determine the number of features to select by themselves, 5-Fold Cross Validation was also implemented for comparison purposes to determine the optimal number of features. Five folds were chosen to ensure that each fold still contain a reasonable number of instances (around 1100), and also to reduce computation time.

The list of number of features selected was k = 10, 20, 30, ..., 120, 132. For each of these k values, the training dataset is separated into 5 folds, where each fold were used as a validation set for a total of 5 iterations. For every iteration, k highest scoring features are selected and the accuracy on the validation set was calculated. The average Accuracy across these iterations were that specific k value's score and used for comparison. This process was repeated for every method and every machine learning (ML) technique, and the highest scoring features across both methods for each technique was be selected for that technique.

Sequential Forward and Backward Selection were also considered as selection methods that determine their own feature number. However, after testing Sequential Forward Selection with 5-Fold Cross Validation and SVM, no improvements in Accuracy were observed (average roughly 45%), plus its significant runtime (roughly 2h30m) meant that these methods were rejected.

2.4 Machine Learning Techniques

2.4.1 SVM

SVMs were chosen as a basic classification method, and a benchmark. Additionally, it was theorised that since all of the features were continuous and numerical, SVMs might be suitable since it aims to draw a hyperplane that separates the classes, and is quite mathematical by nature.

Before being fit to the SVM model, all features were standardised to ensure that features with vastly different ranges and values don't affect the creation of the hyperplanes, since this process involves distance calculation.

2.4.2 RF

RFs were chosen due to them being a good ML method in general. They are usually quite resistant to noise and outliers in data, which are quite possible given the large dataset and range of features. Additionally, RFs also perform their own feature importance ranking, and can be useful when dealing with extracted features that might be less useful.

Features were not scaled or standardised when being fit to RF, since it is a tree-based model and regardless of the scale or range of the features, the data partition process would not be affected.

2.4.3 CNN

Finally, CNNs is a technique that is designed for, and thus is very successful with image classification tasks. CNNs perform their own feature extraction, which generally performs better than self-extracted features. Therefore, it was determined to be quite suitable for the task explored in this report.

Before being fit to the CNN, data images were resized to 25 by 25 to ensure uniformity and consistency in kernel traversal and pooling operations. 25px was chosen because it's the smallest width and height among all images in the training dataset.

Layer	Parameters
Convolutional	filters=25, ReLU
Max Pooling	pool size=2, strides=2
Convolutional	filters=50, ReLU
Max Pooling	pool size=2, strides=2
Flatten	
Dense	units=100, ReLU
Dropout	0.5
Dense	units=43, softmax

Table 1: Implemented layers of the CNN (using Keras and Tensorflow). All kernel sizes are (3, 3).

The final architecture of the CNN constructed in this report can be seen in Table 1. While one Dropout layer was added to help with potential overfitting issues, the remaining layer combination was reached through a process of trial and error, where performance on validation set was evaluated.

3 Results

3.1 SVM

From Figure 1 and Figure 2, we can clearly see that in both metrics, 10 features were the best number to be selected. From 20 onwards, the Accuracy score dropped significantly to around 5%. Additionally, the 10 features selected using the former method performed better than the latter on the validation set. Thus, these 10 features (they were hog_pca 0 to 5, 7 to 9, and 12 for all 5 folds) were chosen for the final model.

The Accuracy for this model on the test set was 33.033%.

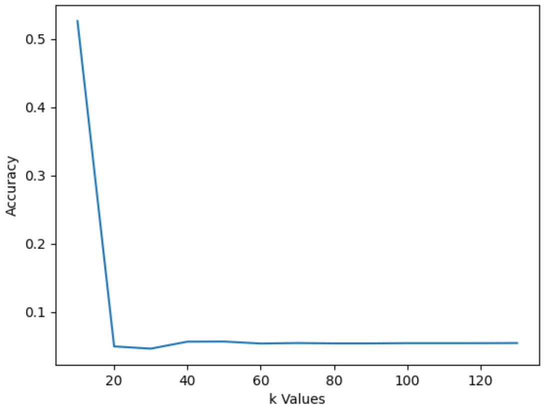

Figure 1: Accuracy score of SVM across different number of features selected using the ANOVA F-value metric.

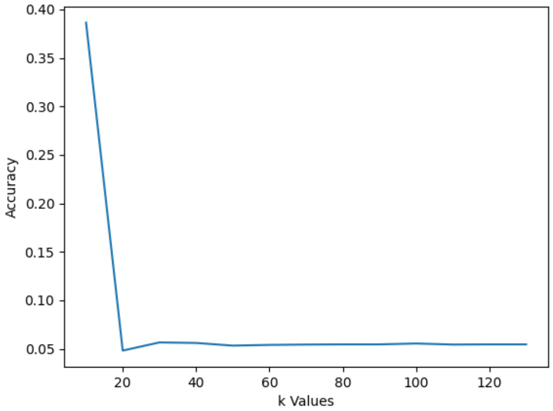

Figure 2: Accuracy score of SVM across different number of features selected using the Mutual Information metric.

3.2 RF

From Figure 3 and Figure 4, it can be observed that in both metrics, the Accuracy score was highest when all features were selected. As a general trend, it seems that as the number of features selected increased, so did the Accuracy score when using RFs. Thus, all features were chosen for the final model.

The Accuracy for this model on the test set was 55.155%.

3.3 CNN

Across different combination and addition of layers to the CNN, the Accuracy score on the validation set ranges from around 89% to 96%. And for all of these models, after a certain number of epochs, their performances plateaus and fluctuates around a percentage. Specifically, for the final model chosen described in Table 1, the Accuracy stopped increasing meaningfully at around epoch 28 or 29, staying around high 95%, close to 96%. Thus, the weights generated after 30 epochs was chosen for the final model.

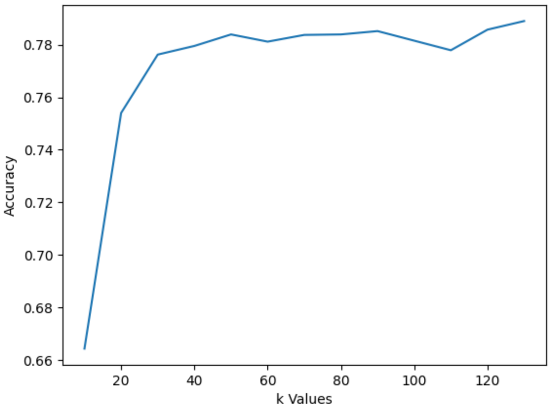

Figure 3: Accuracy score of RF across different number of features selected using the ANOVA F-value metric.

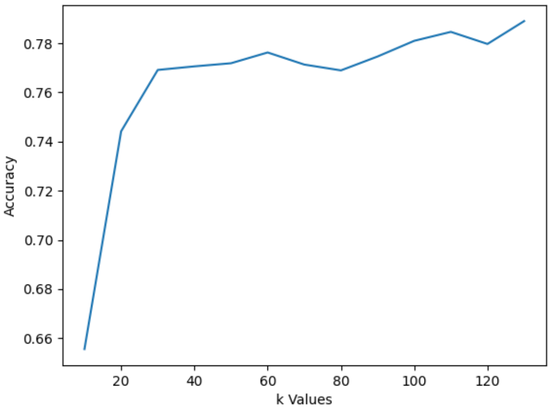

Figure 4: Accuracy score of RF across different number of features selected using the Mutual Information metric.

The Accuracy for this model on the test set was 95.295%.

4 Discussion

4.1 Feature Selection

It can be seen that the inclusion of more features proved to be harmful to the performance of SVMs. This might be due to the fact that SVMs are often more negatively affected by noise and outliers, as the construction of a hyperplane can change drastically if there happen to be, for example, a member of class A that is very close to other members of class B. Therefore, using more features might mean using more bad features that contain a lot of noise. In contrast, it seems that RFs benefited from having more features being used. As RFs construct many decision trees before deciding on a prediction, noise in datasets are unlikely to influence the results of all the trees, which can explain why it wasn't as affected by the features. Additionally, like mentioned, RFs perform their own feature importance ranking, and thus might benefit from having more features.

We also saw that Histogram of Gradients (HOG) features were favored by the feature selection process for SVMs. While other features like mean RGB values or colour histogram can be general descriptors of an image, HOGs are often used specifically for object detection tasks. Not only that, HOGs are robust to changes in illumination and shadowing, which can be helpful as some images in the dataset have quite drastic lighting (see Figures 5 and 6 for examples).

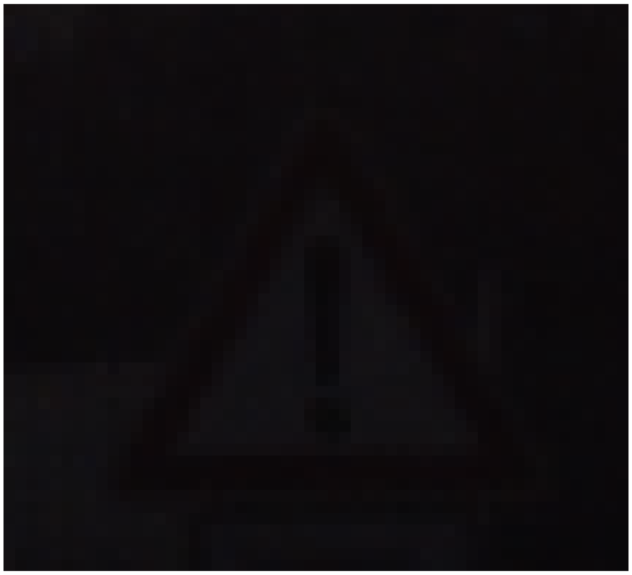

Figure 5: img000011.jpg from the training dataset.

4.2 Performance

It is quite clear that out of the 3 ML techniques explored in this report, CNN was the best performing one, followed by RF, then SVM. This should be expected, since while RF and SVM are more general classification methods, CNN is a very specific method that excels at image classification. Additionally, since CNNs extract their own features, it was not affected by any potential issues or inaccuracies that stem from feature extraction. Moreover, it is also quite reasonable that RF outperformed SVM. SVMs are more suited when handling linear relationships, and can struggle with high-dimensional data. However, the data used in this report likely contains far more complex relationships, and thus is better for RFs. Additionally, the fact that there are 43 classes meant that SVM had to perform a One-vs-Rest approach for classification, which can introduce further errors.

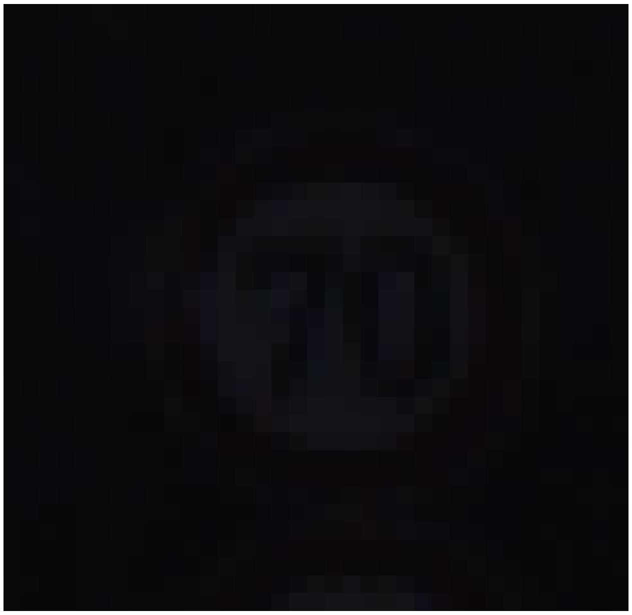

Figure 6: img000090.jpg from the training dataset.

4.3 Other Limitations

Other potential reasons for poor performance include the low resolution and poor image quality of the dataset, as well as the mentioned lighting conditions. From Figure 7, we can also see the uneven distribution of class labels in the training dataset, which might have been another contributing factor.

Improvements to extracted features are also possible, since the detected contours had varying degrees of success. Figure 8 show an example of a good contour detection, while Figure 9 depicts the opposite. These inaccuracies could have led to less useful features, which in turn negatively affected performance.

Finally, other potential features could have been explored, such as ones related to text detection or colour separation, both of which are quite notable characteristics of traffic signs.

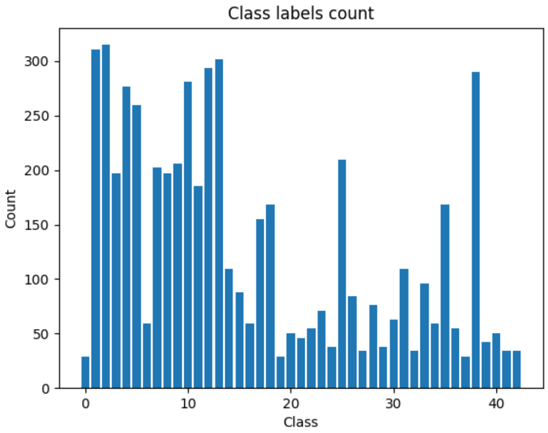

Figure 7: Count of each class label from the training dataset.

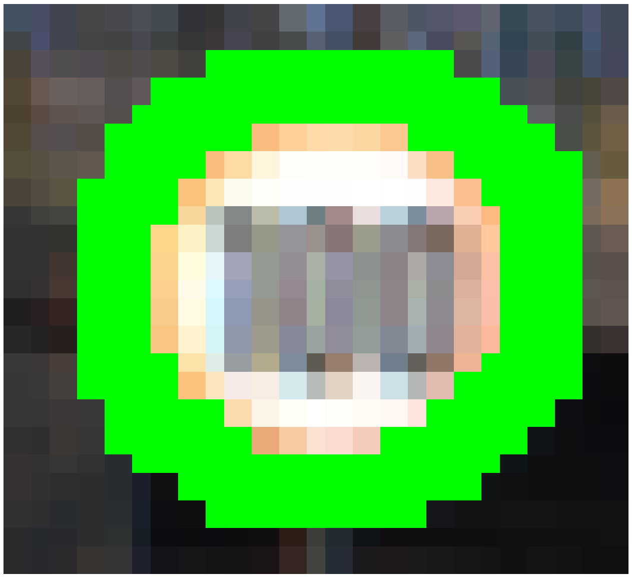

Figure 8: Detected contour (in green) drawn over img_000003.png from the training dataset.

Text detection was briefly explored (with Pytesseract), but not to much success, and was discarded. But further changes could potentially have been made to improve it.

5 Conclusions

In conclusion, exploration into the task of traffic sign classification revealed that the nature of the task, which includes aspects like the characteristics or relationship of the data plays a major role in the success of a ML technique. Additionally, it seems to be the case that CNNs that extract its own features from data tend to outperform more traditional classification techniques using extracted shape-based features.

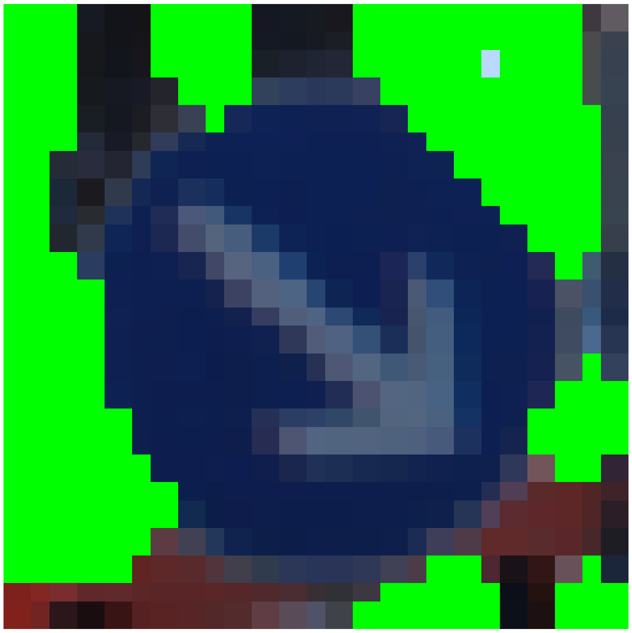

Figure 9: Detected contour (in green) drawn over img_000015.png from the training dataset.

References

Ming-Kuei Hu. 1962. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187.

Nobuyuki Otsu. 1979. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66.