Image Classification of Traffic Road Signs Using Machine Learning

Team 33

Lim Zeen Kiat, Kevin Tan Shin Wei, Robin Ghosh, Delphino Chew Jing Cheng

Introduction

What was once considered a pipe dream, has slowly come within reach over the past few decades. Autonomous cars, also known as self-driving cars, are vehicles that can operate and drive themselves without any human assistance. As technologies like artificial intelligence continue to develop at breakneck speeds, fully autonomous cars have become more plausible, threatening to change the landscape of transport completely. Such cars have already proven to be capable of basic functions like braking and steering (Garsten, 2024), allowing them to drive in controlled conditions. They can move and stop appropriately, complete turns and even navigate simple traffic congestion. Unfortunately, there are still many complex aspects of driving that must be resolved before self-driving cars are fully autonomous and functional.

One aspect is road regulations. Traffic regulations conveyed through traffic signs, unique to each traffic rule, must be read and comprehended by the autonomous vehicles for safe and orderly operations. Thus, a fully autonomous car must be able to identify road signs and ascertain their meaning as it drives.

A logical way to do this would be to use the images of the road signs captured by the autonomous car's cameras and classify them before responding according to the class's rules. In this project, we aim to develop a model that classifies traffic sign images accurately. We will compare 3 different models used for image classification, comprising of both machine learning and deep learning models, and determine which is the most effective in classifying traffic road signs.

Related Works

There have been several works on different methods for image classification. The 3 models we will be considering are Convolutional Neural Networks (CNN), Support Vector Machine (SVM) and Random Forest (RF).

CNN is a deep learning model that has consistently performed well in image classification. Qiao (2023) developed a CNN deep learning network to detect and recognize traffic signs. After optimization, her model was highly effective and achieved an accuracy of 99.9%. However, the main drawback of CNN is the danger of overfitting (Elngar et al., 2021). This could cause the model's performance to be significantly affected by external factors like lighting, blurriness or the angles of the images, which may distort the images and lead to misclassification.

SVM and RF are 2 popular machine learning algorithms that have also been applied to traffic sign recognition to relative success. Yang Liu's innovative model based on SVM had an accuracy of 90.2% over 15 different road signs (Liu & Zhong, 2020), while Narayana and Bhavani (2022) utilized a Random Forest algorithm that had an accuracy of 86.5%. However, ultimately such traditional machine learning models are typically outperformed by deep learning models given large datasets (Wang et al., 2020). Hence, we aim to develop a CNN model, taking various steps to minimize overfitting, and compare it against SVM and RF models to determine if it is the most accurate.

Dataset

The dataset consists of 4896 images of road signs, comprising of 23 distinct types. Each type of road sign contains image samples ranging from 100 to over 500. These are Red-Green-Blue (RGB) images which are represented by pixels. Each pixel is represented by 3 different values from 0 to 255, with each value being the intensity of the 3 colours. Unfortunately, different images have varying sizes, namely their height and width. Thus, before we can utilize this data, we must first process it to suit our needs.

Firstly, we perform feature scaling on the dataset by normalizing it. This is done by dividing all the values by the maximum colour value of 255, scaling the pixel values to the range of 0 to 1. Next, to standardize the sizes of the images, we have decided to scale them to a size of 94 x 94 pixels, optimizing for computation cost and precision.

For Support Vector Machine and Random Trees, we flattened each image into a 1-dimensional (1D) array where each of the red, green and blue values per pixel are also separated into individual features. Those models do not have spatial awareness and hence require a 1D array of features per image as input. Then a 2-dimensional (2D) array is created by joining the 1D arrays by rows for input into the models.

Methods

Approach

Our approach consists of 2 stages, as shown in Figure 1.

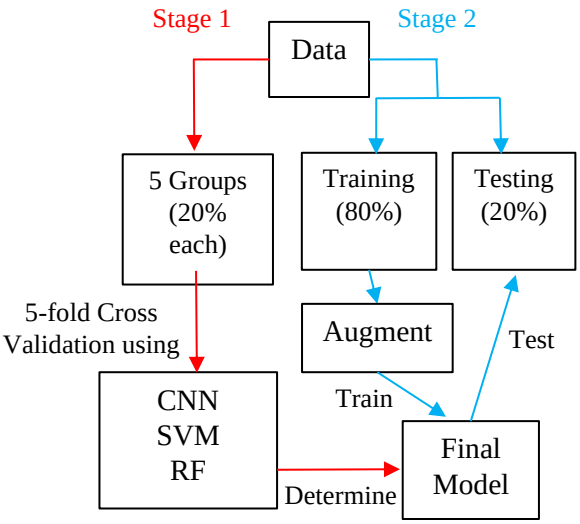

Fig. 1: 2 Stage Approach

We will first use 5-fold cross validation to choose the best model, splitting the data into 5 groups. Each group maintains the proportions of each class within the full dataset. We choose accuracy as our evaluation metric as the dataset is relatively balanced where there is no overwhelming majority of a specific class. Furthermore, in the context of traffic signs, the correct classification of all classes is equally important as each sign has a unique meaning that autonomous cars must consider. Hence, accuracy is the most suitable metric as it judges the model's overall correctness rather than the predictions of specific classes.

Secondly, we will create and evaluate the chosen model. The original dataset will be randomly split into training and testing data, 80:20 respectively (80-20 split), while maintaining the original class proportions. The training data will be augmented with randomized Gaussian noise and blurring, before training our model. This ensures that our model will be robust to small variations in the input data, allowing it to generalize better and hence minimize overfitting. Finally, we will evaluate our model on the testing data to determine its accuracy.

Before that, we will first explain how the 3 models work and how we implemented and fine-tuned them.

Convolutional Neural Network

Convolutional Neural Network (CNN) is a supervised deep learning model which is used for image pattern recognition. The model accepts a 3D array of normalized pixel values of images from our dataset. The input is then processed through multiple layers or neural networks such as convolutional layers, pooling layers and fully connected layers. Important features are extracted in each layer, allowing the model to learn spatial hierarchies of patterns.

To determine which model framework will be most suitable for our dataset, we used a pretraining scheme called transfer learning. Transfer learning uses models trained on other large datasets as a foundation to build our model (Krishna & Kalluri, 2019). We tested the MobileNetV3, ResNet50 and VGG16 individually as suitors for our base model.

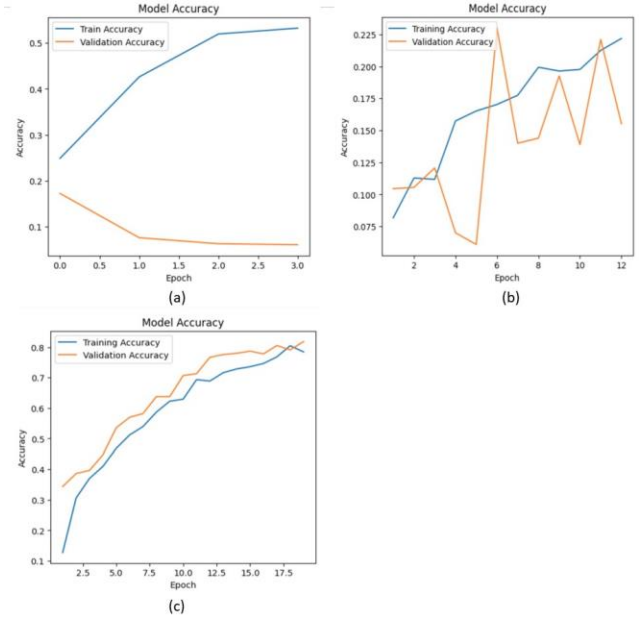

Fig. 2: Graphs of accuracy plotted against epoch for models using (a) MobileNetV3, (b) ResNet50, and (c) VGG16, as our base model

From the results in Figure 2, the VGG16's framework is the most suitable for our dataset. However, it is too computationally expensive to use as our base model. Therefore, we created a simpler model, similar to it. To reduce the chance of overfitting, we added dropout layers, which randomly disable some neurons during training to prevent over-reliance on certain neurons (Nagyfi, 2018). In the output layer, we used a "softmax" activation function with 23 neurons. The model then produces a probability distribution across 23 classes and classifies each image according to the class with the highest probability. Finally, we used Adaptive Moment Estimation (Adam) as our optimizer because it has adaptive learning rates and utilizes bias-correction mechanisms, ensuring computationally efficiency and stability of the training, which is suitable for image classification with noisy and diverse features (Ruder, 2017). We also used a small learning rate of 0.0001 to improve generalization and

sparse categorical cross-entropy as our loss function since it is designed for multi-class classification.

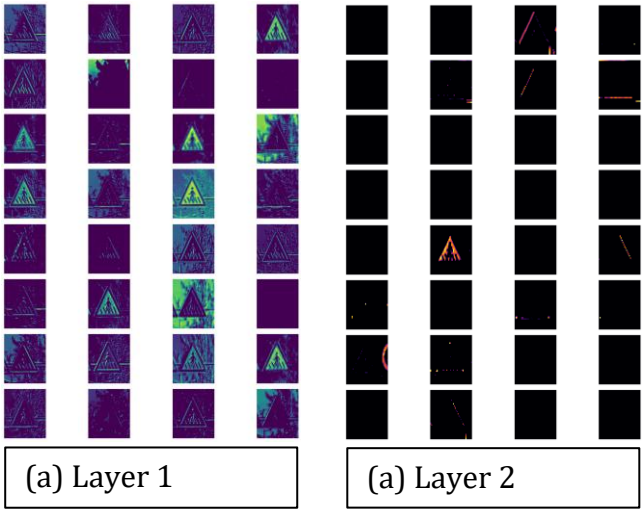

Fig. 3: Feature Visualization for a Conv2D layer

Our CNN model has two Conv2D layers which scan small portions of each image to detect its local features. Fig 3 shows a visualization of the features detected by each layer. Each picture from a layer is the visualization of features detected by a single neuron in the layer. The bright parts of the picture indicate the focus of that neuron. Studying the collage of pictures, we can see that each neuron detects a specific trait – an edge, shape or colour. Collectively, these neurons produce results that distinguish each class for the deeper layers to process.

Support Vector Machine

In Support Vector Machines (SVM), hyperplanes are used as decision boundaries to separate each class in the dataset. The model takes in a 2D array of normalized pixel values of our image training dataset. The model determines the optimal hyperplane for each class that maximizes the distance between the class's data points and the others (Pisner & Schnyer, 2019).

As the decision boundaries for each class in the traffic sign images is highly non-linear, we needed a flexible kernel to capture these complex patterns. Hence, the Radial Basis Function (RBF) kernel was used which maps the data to an infinite-dimensional space. A penalty parameter of 1.0 was chosen to balance training accuracy with generalization to the test set, avoiding overfitting. Each test sample's class is voted by checking its position relative to each decision boundary, then the class that appears the most is chosen.

Random Forest

Random Forest (RF) is a supervised machine learning algorithm which classifies using decision trees. The model works by constructing multiple decision trees and combining their outputs. Like SVM, the model accepts a 2D array of normalized pixel values of the images in the dataset. From the training dataset, RF selects multiple subsets of the original data, where each is used to create corresponding unique decision trees ensuring that no tree is highly influenced by the entire dataset - termed as overfitting. Each tree classifies the images by splitting the data at each level based on learned factors like pixel intensity values.

We implemented the model using a maximum tree depth of 8 concurrently with an ensemble of 100 trees. A shallow tree depth of 8 prevents each tree from memorizing its dataset. The large number of trees creates greater diversity among the trees, allowing for greater generalization.

Each tree makes a prediction for a given image and votes for a class. The final classification depends on the majority vote of all the trees. Random Forest can handle high dimensional datasets as each tree looks at a random subset of the pixel values of each training sample to extract features for its decisions. This also improves generalization of the model. The model, however, lacks the spatial awareness of deep learning models, limiting its ability to only capturing the overall structure of the images.

Results and Discussion

After training and testing the 3 models using 5-fold cross validation, Figure 3 shows the accuracy of each model per iteration, and its average accuracy.

	CNN	SVM	RF
Iteration 1	98.40	94.22	94.83
Iteration 2	99.03	95.02	93.39
Iteration 3	99.97	96.43	94.70
Iteration 4	99.97	93.95	92.72
Iteration 5	99.21	95.88	94.12
Average	99.32	95.10	93.95

Fig. 3: Accuracy in % for each model using 5-fold cross validation

CNN is clearly the best performing model for the classification of traffic signs, with an average accuracy of 99.32% (Fig. 3). Thus, we will use CNN as our final model and evaluate it.

Evaluating Final Model

We then create our final model using CNN. We train it using the augmented data before evaluating it using the remaining test data.

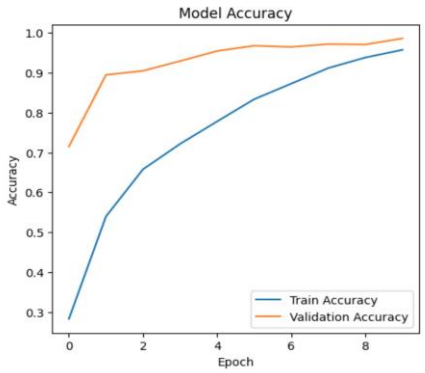

Fig. 4: Graph of model accuracy w.r.t epoch

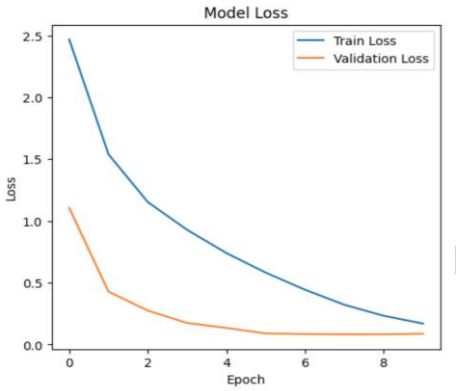

Fig. 5: Graph of model loss w.r.t epoch

From Figures 4 and 5, we can see that the validation accuracy and loss continually improve, implying that our model continues to perform well on unseen data and is not overfitting. While it is not ideal that the model underperforms during training in terms of both accuracy and loss compared to during validation, this is a side effect of adding dropout layers which reduces training accuracy (Chollet et Al, 2015) and is necessary to reduce overfitting. Overall, our CNN model performs extremely well, with a remarkable test accuracy of 97% and test loss of 0.08.

Conclusion

In conclusion, all 3 models, CNN, SVM and RF generally perform well and produce a high accuracy of above 90%, which is crucial for the task of traffic sign image classification. However, the CNN model stands out because it can classify images based on local features instead of the entire image. Traffic signs may be encountered from different angles, under different conditions, making it more important to recognize the signs based on the shapes, edges and colours present instead of the full, ideal image. Hence, CNN fits the best for this purpose.

With the ability to correctly classify traffic signs, autonomous cars are becoming more and more of a reality. However, our model is not quite fully there yet. Our model, while highly accurate, misclassified some signs as another sign with similar shapes or colours, such as a red circle or a yellow triangle, as shown in Figure 6.

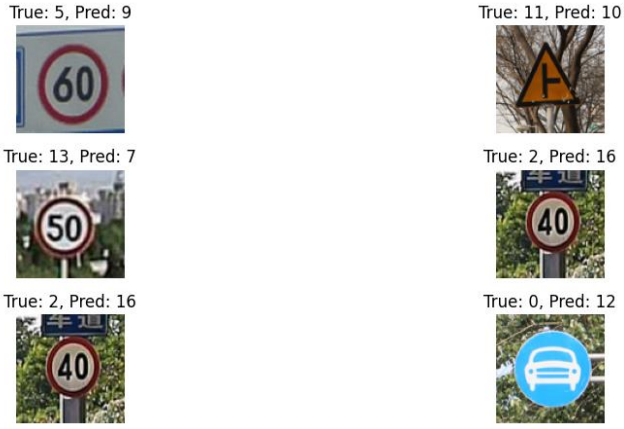

Figure 6: Examples of misclassified images

This shows that our model still needs to improve by prioritizing the features within the sign itself. With road safety being of utmost importance, such errors are unacceptable, and the car's ability to classify road signs needs to be just as good as that of a human, if not better, before they can be fully autonomous.

References

Chollet, F. et al (2015). *Keras FAQ*. Keras. https://keras.io/getting_started/faq/#why-is-my-training-loss-much-higher-than-my-testing-loss

Elngar, A. A., Artificial Intelligenc, Arafa, M., Fathy, A., Moustafa, B., Mahmoud, O., Shaban, M., & Fawzy, N. (2021). *Image Classification Based On CNN: A Survey*. https://doi.org/10.5281/zenodo.4897990

Garsten, E. (2024, January 23). *What Are Self-Driving Cars? the Technology Explained*. Forbes. https://www.forbes.com/sites/technology/article/self-driving-cars/

Krishna, S. T. K., & Kalluri, H. K. (2019). Deep learning and transfer learning approaches for image classification. *International Journal of Recent Technology and Engineering (IJRTE)*, *7*(5S4), E10900275S419. https://www.re-

searchgate.net/profile/Hemantha-Kumar-Kalluri/publication/333666150_Deep_Learning_and_Transfer_Learning_Approaches_for_Image_Classification/links/5cfbeeb9a6fdccd1308d6aae/Deep-Learning-and-Transfer-Learning-Approaches-for-Image-Classification.pdf

Liu, Y., & Zhong, W. (2022). A Novel SVM Network Using HOG Feature for Prohibition Traffic Sign Recognition. *Wireless Communications and Mobile Computing*, *2022*, 1–14. https://doi.org/10.1155/2022/6942940

Nagyfi, R. (2018, December 3). The differences between Artificial and Biological Neural Networks. *Medium*. https://towardsdatascience.com/the-differences-between-artificial-and-biological-neural-networks-a8b46db828b7

Narayana, M., & Bhavani, N. P. G. (2022, February 1). *Detection of traffic signs under various conditions using Random Forest algorithm comparison with KNN and SVM*. IEEE Xplore. https://doi.org/10.1109/ICBATS54253.2022.9759067

Pisner, D. A., & Schnyer, D. M. (2019). Support vector machine. Machine Learning: Methods and Applications to Brain Disorders (pp. 101–121), in *Elsevier ebooks*, from https://doi.org/10.1016/b978-0-12-815739-8.00006-7

Qiao, X. (2023). Research on Traffic sign recognition based on CNN Deep Learning Network. *Procedia Computer Science*, *228*, 826–837. https://doi.org/10.1016/j.procs.2023.11.102

Ruder, S. (2017). *An overview of gradient descent optimization algorithms* (arXiv:1609.04747v2). https://arxiv.org/pdf/1609.04747

Wang, P., Fan, E., & Wang, P. (2020). Comparative Analysis of Image Classification Algorithms Based on Traditional Machine Learning and Deep Learning. *Pattern Recognition Letters*, *141*. https://doi.org/10.1016/j.patrec.2020.07.042