Traffic Sign Images Classification with Deep Learning

Team 3
Evan Ng, Kyi Wai, Mario Hanzel, Sydney Sim

Introduction

Image classification is a crucial area that enhances the efficiency of various systems. Fatal traffic accidents surged by 26% from 2022 to 2023, surpassing pre-Covid levels, increasing urgency towards road safety improvement (SPF 2023). Traffic signs provide critical information to drivers. Recent advancements in computer vision and artificial intelligence have made traffic sign classification a vital research focus with significant implications for traffic safety.

Fig 1. Usage of Computer Vision in Autonomous Driving Systems

The growing use of autonomous vehicles has increased the demand for reliable traffic sign recognition and classification models, as their accuracy is crucial for ensuring driver and passenger safety. Many modern vehicles already incorporate these technologies to assist drivers with real-time sign interpretation. Expanded research and accessibility of these systems could enhance safety standards in the automotive industry. This project explores the key components needed to create a robust traffic sign recognition model.

Related Works

A wide variety of techniques exist to classify images on the road, from Support Vector Machines (SVM), Decision Trees and Random Forests. This paper examines recent developments in traffic sign classification, reviews the existing benchmarks and techniques, and explores potential pathways to improve accuracy and reliability in real-world applications. One study conducted in 2011 using SVMs to classify traffic signs achieved roughly 98% accuracy on different sets of split data (Dong 2021). One key drawback of this method is that the researchers had to split each class further into super classes before using the algorithm. As a result, this specific method of multi-class SVM classification is not generalisable and is context dependent. Another paper analyzed the performance of a custom CNN (Convolutional Neural Network) against other CNNs architectures, finding the performance of the custom model to have an accuracy of 99.81% (Deepika et al. 2023). Many of the drawbacks mentioned earlier are indeed solved by CNNs, with many studies achieving above 95% accuracy on the GTSRB dataset. Given that CNNs seem to be the model that would achieve best results and be the most generalizable, we have chosen CNNs, with varying architectures and schemes for this task.

Dataset

The training dataset includes 23 distinct classes of traffic signs, using a 70-20-10 train-validation-test ratio. Our analysis revealed an imbalance, with lower class indices representing a larger portion of the data (see Fig. 2). This highlights the need to ensure fair representation of all classes to prevent bias towards class 0 in our model.

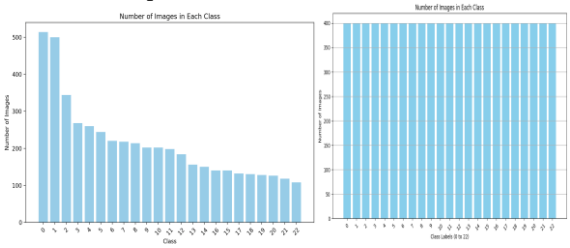

Fig 2. The Distribution of classes

Furthermore, we realized that some signs are more important than others when it comes to real traffic scenarios. Incorrect recognition of such signs (Fig 3.) can lead to accidents. These signs are identified by us as more "important" and are thus our positive classes for input.

Fig 3. Pedestrian and Stop sign

Data Augmentation

Given a relatively small-moderate dataset that differs from many benchmark traffic classification datasets such as the GTSRB by one order of magnitude, the data may not be sufficient to train a robust model. To combat this, we augment the pre-existing data, doing a series of rotations and transforms. This would ensure that the model is not overfitted on the simple dataset, where the test data is almost the same as the training data.

To achieve uniform distribution amongst the classes, we under sampled dominant classes to the median number of images, 198. Then, we oversampled all classes to achieve a target of 400 images per class. This augmented dataset comprises random rotations, random zooms, adjustments to brightness, and the introduction of Gaussian noise, all of which simulate various conditions encountered on roads (see Fig. 5). We had to be meticulous when choosing augmentations, given that we want a generalizable model. We avoid flipping images vertically or horizontally, as certain signs (Fig. 4) could train the model to misinterpret orientations, which is especially critical in self-driving systems. This approach helps the model correctly recognize traffic signs in various conditions. Additionally, we standardized the pixel values by dividing each pixel by 255, which facilitates faster computations. Only after these steps have been taken will the data be ready for use in the model.

Fig 4. Horizontally flipped left turn sign

Fig 5. Street Signs After Augmentation

Setting Hyperparameters

We chose the Adam optimizer over SGD because it is able to automatically adjust the learning rate for each parameter individually, while SGD adjusts for all parameters. This is crucial to combat potential bad parameter initialization, and generally takes the guesswork out of model fine tuning. A study done indicated that a batch size of 256 and learning rate of 0.001 resulted in the most accurate results for the Adam optimizer (Temraz and Keane 2022). We use early stopping conditions with patience of 5 epochs and a learning rate decay factor of 0.5 with patience of 2 epochs, which will enable us to find the loss function minima with minimal manual input.

Methods

We experimented on 5 different models: custom CNN model with a single convolutional layer, another custom CNN model with more layers, ResNet50 pretrained model, pretrained MobileNetV2 model without fine-tuning, and MobileNetV2 model with fine-tuning.

Before we started this experiment, we wanted to find out if getting a better, more accurate model was as easy as increasing model depth/complexity. It seemed almost intuitive that with more complex models, we would be able to capture more complex, non-linear patterns that otherwise would not have been captured by a simpler model.

Our first approach to the problem was the simplest one, using CNN, to try to classify the images. CNNs work by applying convolution filters to detect features in the input data. Different filter layers detect specific local features, such as edges or textures. Stacking these layers allows the network to learn complex and abstract features (refer to Fig. 6).

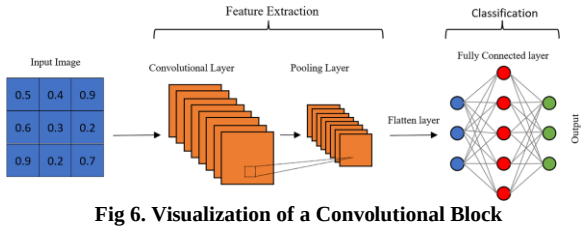

Fig 6. Visualization of a Convolutional Block

The first model is a custom CNN with one singular convolutional and pooling layer, followed by fully connected layers. This model was chosen to establish a baseline by training CNN from scratch.

The second model is an enhanced custom CNN featuring three convolutional and pooling layers. This design increases the model's capacity to capture intricate patterns in the image data.

For our third model, we used a ResNet50 pre-trained on ImageNet, leveraging the model's learned features. ResNet50 is a CNN architecture that belongs to the ResNet (Residual Networks) family and has a total of 50 layers (refer to Fig 7). As Ibrahim (2024) explains, ResNet50 has a residual block which allows some layers to be skipped. Hence, this enables the gradients to flow more smoothly, addressing the vanishing gradient problem.

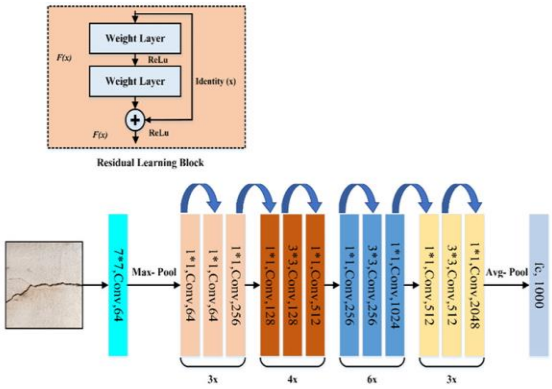

Fig 7. Visualization of Resnet50

The fourth model, MobileNetV2, is a type of CNN specifically designed to work efficiently on mobile and resource constraint environments (Sandler et al. 2018). Hence, it is created to use less power and memory than larger models. Sandler et al. (2018) highlights that MobileNetV2 architecture uses techniques like depthwise separable convolutions, inverted residuals, and bottleneck layers to reduce computational cost while maintaining accuracy (refer to Fig 8.). We added an output layer to the MobileNetV2 with a SoftMax activation function to classify into 23 categories. We trained just the output layer to update the weights of this layer to allow the model to correctly classify the traffic sign images.

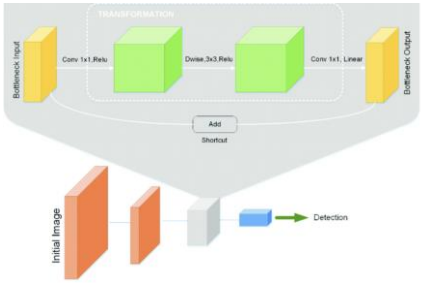

Fig 8. Visualization of MobileNetV2

For our last model, we fine-tuned the fourth model by unfreezing the last 5 layers of the ModelNetV2 model, allowing the model to update these layers to adapt more closely to our datasets. This fine-tuning aimed to enable the model to capture specific features relevant to our 23-category traffic images classification task.

Results

To assess the effectiveness of our model, we concentrated on evaluating its accuracy, false negative rate, and F1 score. We chose to prioritize the F1 score, as it is essential for our model to maintain both a low false negative and false positive rate. Emphasis was placed on minimizing the false negative rate for critical traffic signs, such as stop signs and pedestrian crossing signs, since misinterpretation of these could result in severe consequences. The results of our testing are as follows (refer to Fig. 9). Our testing shows that, even though model complexity does improve model performance, this is only to a certain extent.

Model	Test Accuracy	F1 score (class 6, stop-sign)	F1 score (class 10, pedestrian crossing sign)	Total Misclassifications
ResNEt 50	0.56	0.73	0.567	328
Shallow CNN	0.87	0.90	0.96	95
Deep CNN	0.97	0.98	1.0	26
Mobile NetV2	0.98	1.0	1.0	12
Fine Tuned MobileNet V2	1.00	1.0	1.0	2

Fig 9. Test Results

The accuracy of the models vs training epochs are plotted below for your reference (refer to Fig 10). Note that these models were trained until they plateau in validation loss.

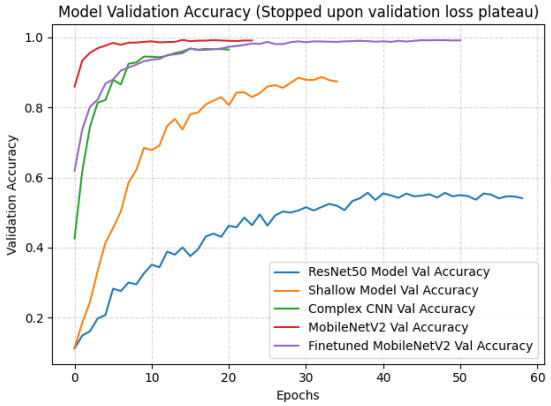

Fig 10. Model Validation Accuracy

Despite the advantages of the ResNet50 architecture as mentioned above, in our experiments, ResNet50's performance on this dataset was subpar as mentioned earlier. This could be due to the relatively small dataset size of 9,200 images (even after data augmentation), which may not provide enough diverse examples for a large model with many layers like ResNet50 to generalise well. The high number of deep layers in ResNet50 may make it prone to overfitting.

Discussions

The results reveal that model complexity can significantly influence performance. While a deeper CNN model

improved accuracy compared to a simple shallow CNN, further increasing complexity by using ResNet50 resulted in a sharp decline in performance. This suggests that our dataset might not be sufficiently large or diverse to take advantage of deeper architectures like ResNet50, which typically require extensive training data to generalise well.

A noteworthy observation is the outstanding performance of MobileNetV2, especially after fine-tuning. MobileNetV2 is specifically designed for efficiency and works well with smaller datasets, maintaining high accuracy with fewer computational resources. Fine-tuning the model by unfreezing some of its layers allowed us to better capture the nuances of our traffic sign data, leading to almost perfect performance metrics.

The contrast in performance between ResNet50 and MobileNetV2 also highlights the importance of selecting an appropriate model architecture based on the size and complexity of the dataset. The poor performance of ResNet50 could be due to insufficient training data, which hinders the model's ability to learn meaningful features. This emphasises the need to match the model's complexity to the available data.

In future work, collecting a more extensive and diverse dataset or using transfer learning with more data-specific pre-training could be explored to potentially enhance performance further.

Another crucial aspect to consider is that traffic signs in real-world scenarios may be affected by various environmental factors, such as streaks of light, harsh shadows, or even vandalism, all of which could sharply decrease the model's accuracy (Cao et al. 2024). Addressing these challenges might require further augmentations or the development of models capable of handling such variations to ensure robust and reliable performance.

Reference

Alsaleh, A.; and Perkgoz, C. 2023. A Space and Time Efficient Convolutional Neural Network for Age Group Estimation from Facial Images. *PeerJ Computer Science* 9 . doi.org/10.7717/peerj-cs.1395

Cao, H.; Yuan, L.; Xu, G.; He, Z.; Fang, Z.; and Fang, Y. 2024. Secure Traffic Sign Recognition: An Attention-Enabled Universal Image Inpainting Mechanism Against Light Patch Attacks. *arXiv preprint*. arXiv:2409.04133.

Deepika, V.; Vashisth, S.; and Sharma, P. 2023. An Efficient Traffic Sign Classification and Recognition with Deep Convolutional Neural Networks. International Journal of Convergence in Healthcare 3(2). Doi.org/10.55487/p9zr8384.

Kandel, I.; and Castelli, M. 2020. The Effect of Batch Size on the Generalizability of the Convolutional Neural Networks on a Histopathology Dataset. *ICT Express* 6(4): 312–315. doi.org/10.1016/j.icte.2020.04.010.

Land Transport Authority (LTA). 2023. *Statistics*. Singapore: Land Transport Authority. https://www.lta.gov.sg/content/ltagov/en/who_we_are/statistics_and_publications/statistics.html.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4510–4520. doi.org/10.48550/arXiv.1801.04381.

Singapore Police Force. 2023. *Annual Road Traffic Situation 2023*. Singapore: Singapore Police Force. https://www.police.gov.sg/-/media/D4435F72157942D3B323EE4A507D4CFB.ashx.

Zaklouta, F.; Stanciulescu, B.; and Hamdoun, O. 2011. Traffic Sign Classification Using K-d Trees and Random Forests. In *Proceedings of the 2011 International Joint Conference on Neural Networks*, 2151–2155. San Jose, CA: IEEE. doi.org/10.1109/IJCNN.2011.6033494.