Convolutional Neural Network Transfer Learning with VGG16 to Classify Traffic Signs

Amy Reidy, MSc. In Computing (Data Science)
Institute of Technology Sligo

Abstract

The aim of this project was to create a classification model, using a pre-trained convolutional neural network, that can effectively classify images from the German Traffic Sign Recognition Benchmark dataset. The convolutional base of the popular VGG16 model was used for feature extraction, while the top fully connected layers were retrained with the GTSRB dataset, and the hyperparameters were fine-tuned to further improve the model. The results show that the best performing model in this experiment achieved an accuracy of 82.98%. As the dataset is quite imbalanced, future work could improve on this result by augmenting the images in the smaller classes to create a more balanced dataset.

Introduction

This project uses transfer learning with a pre-trained convolutional neural network (CNN) to create a model that can accurately classify traffic road signs. Creating a CNN from scratch requires a lot of computational power and time, and it needs to be trained on a very large dataset to reduce overfitting. Whereas, transfer learning is a machine learning method that allows us to use a pre-existing deep learning model to quickly classify a similar smaller dataset. This project utilizes a very popular state of the art image classification model called VGG16 that was proposed by Simonyan and Zisserman (2014).

Data

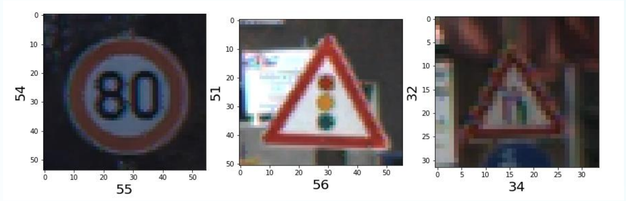

Fig 1. Examples of Images in Dataset

- The images used for this project come from the German Traffic Sign Recognition Benchmark dataset that was developed for an image classification challenge at the International Joint Conference on Neural Networks (Stallkamp et al., 2011).
- Over 50,000 images of traffic signs (including over 12,500 testing images), and 43 imbalanced classes of traffic signs (see Figure 1).
- The images were resized to 224 x 224 pixels, converted to BGR and each colour channel was zero-centered with respect to ImageNet (the dataset that VGG16 was originally trained with).

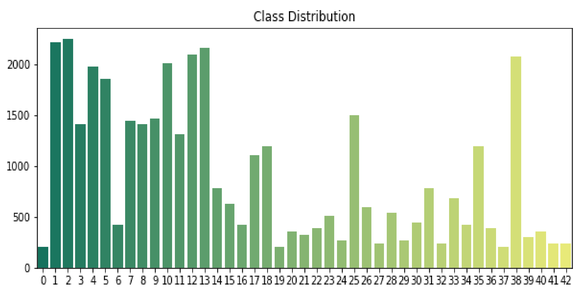

Fig 2. Plot of Imbalanced Classes

Methods

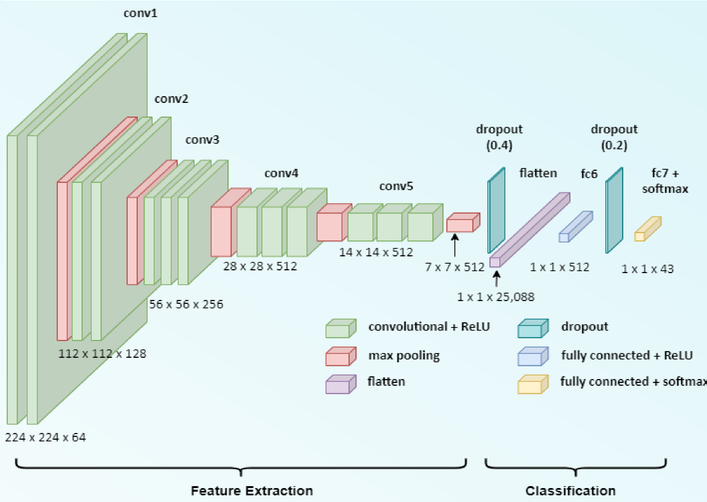

Fig 3. Network Architecture of Model E

- The full VGG16 network has 13 conv. layers with 5 max pooling layers, and 3 fully connected layers at the end. The conv. layers use 3 x 3 filters with a stride of 1 and the max pooling is carried out by a 2 x 2 filter with a stride of 2. Only the conv. base of VGG16 was used in this project and the conv. layers were frozen to retain their original weights.
- After the conv. blocks, the data was flattened into a one-dimensional array, then fed into a fully connected layer with 512 neurons and the ReLU activation function was applied.
- The Softmax activation function was applied to the final layer to output probabilistic values for each of the 43 classes.
- Categorical cross entropy was used as the loss function, and all models were optimized with the Adam algorithm.
- Models were first tuned by trialling different batch sizes (256 was found to be most effective) and then further improved by varying the learning rates and adding dropout layers and class weights to reduce overfitting.

Results

- Model E was the top performing model (see Table 1), and it achieved a test accuracy of 82.98% and a test loss of 0.8987. This model's architecture is shown in Figure 3.
- Figure 4 shows the accuracy and loss results during training and validation of Model E. As early stopping checkpoints were used, training stopped after epoch 28 and the weights from epoch 25 were restored as the validation loss started to increase after this iteration.
- The training loss for Model E was quite high for the early epochs, more so than in other models, and this is likely due to the addition of dropout layers and class weights.

	Learning Rate	Dropout	Class Weights	Val Loss	Val Accuracy	Test Loss	Test Accuracy	Run Time
A	0.001	No	No	0.083	0.9754	0.9619	0.7972	7min 44s
B	0.001	1 layer	No	0.0904	0.9709	1.0602	0.8095	11min 17s
C	0.001	2 layers	No	0.0762	0.9786	1.1467	0.8249	17min 19s
D	0.0001	2 layers	No	0.0424	0.9885	1.1591	0.8205	35min 56s
E	**0.0001**	**2 layers**	**Yes**	**0.0348**	**0.988**	**0.8987**	**0.8298**	**34min 17s**

Table 1. Results of Top 5 Models

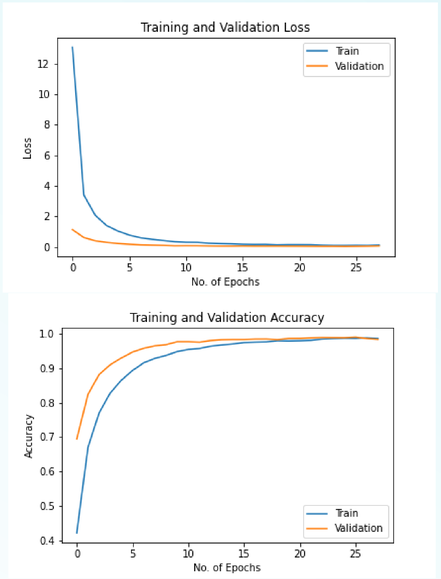

Fig 4. Accuracy and Loss for Model E Training

Discussion

- The results show that quite a high degree of accuracy can be achieved quickly by using transfer learning with VGG16.
- Adding dropout layers slightly increased the accuracy but reducing the learning rate did not; while adding class weights was effective at reducing the loss.
- Even for the top model, there is a significant difference between the performance on the validation set versus the test set, which indicates that the model has overfitted.
- To further improve the model's accuracy, future work could use data augmentation for just the smaller classes to generate more data and thus balance out the dataset. Another alternative would be to try to experiment with retraining some of the convolutional layers.

Conclusion

This project demonstrates how transfer learning can be used with the VGG16 network to classify a smaller dataset quickly and easily. The best model in this project achieved 83% test accuracy, however the difference between the test and validation accuracy suggests that the model has overfitted, and this accuracy is still quite low for a critical task like classifying road signs. And so, it is recommended that future studies apply data augmentation to the minority classes in GTSRB to create a more balanced dataset and a more accurate model.

References

- [Simonyan and Zisserman, 2014] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-scale Image Recognition. arXiv preprint arXiv:1409.1556.
- [Stallkamp et al., 2011] Stallkamp J., Schlipsing M., Salmen J. & Igel C. (2011). The German Traffic Sign Recognition Benchmark: A Multi-class Classification Competition. The 2011 International Joint Conference on Neural Networks, pp. 1453-1460, doi: 10.1109/IJCNN.2011.6033395.

More Information

Please scan the QR code to access the project's source code, accompanying paper and dataset on GitHub.