Student 1: Ngoc Bao Vy Le - s3828276
Student 2: Minh Hoang Nguyen - s3712611

Introduction (¼ page)

There has been a growing research interest in traffic sign recognition (TSR), driven by the growing popularity of autonomous driving vehicles, advanced driver assistance, and mobile mapping in recent years [1]. The traffic sign recognition problem consists of traffic sign localization, to locate traffic signs within a video frame, traffic sign classification, to categorise traffic signs into predefined groups, and real world traffic sign detection [2]. This paper will focus on exploring traffic sign classification and the evaluation of certain neural network models, which could be used to correctly label traffic signs.

The paper will first go through related state of the art works in this field in Sect. 2, providing information about the dataset being used in Sect. 3, and then present the neural network models that were experimented in Sect. 4. Results and the ultimate judgement are addressed in Sect. 5 and 6.

There are two traffic sign dataset being used in this research which are the modified Belgium Traffic Sign Recognition Benchmark (BTSRB) dataset, which were splitted into 80% training set and a 20% performing test set, and a smaller Australian Traffic Sign (ATS) dataset, which was collected from the internet with the main purpose as to evaluate the trained model in a real world scenario. An assumption that the trained models will perform better on the 20% extracted BTSRB test set than on the ATS dataset due to the differences between Australian traffic signs and Belgium traffic signs was made.

The two main classification tasks that these trained models need to achieve are to classify images according to the sign-shape (For example: diamond, hex, rectangle, round, etc) and sign-type (stop, warning, parking, etc).

Related Work (½ page)

(Ciresan et al., 2011) has used MLP and CNNs for traffic sign recognition by further training on the nets which are based on the won the preliminary phrase of German traffic sign recognition benchmark. The CNNs are trained on raw pixels whereas MPL are trained on feature descriptors. The model is developed to classify 43 classes on (48 x 48) pixels . In the result, this MLP/CNNs model has increased the recognition performance rate from 98.98% to 99.15% .

In 2013, Wu et al. developed another traffic sign recognition algorithm which was inspired by CNNs. The method used fixed and learnable layers to detect traffic signs on scene images with the input image range (16x16) to (128x128) and it focused on "prohibitory", "mandatory" and "danger" sentiment. The model also employed the bootstrap method to transfer learning from misclassified train samples to narrow the false positive and false negative classification which lead to the improvement of precision and recall. Besides, to prevent the overfitting problem, data augmentation of intensifying positive samples by transformation (rotation, translation and scale). The model had achieved a good result with 99.73% and 97.62% accuracy in the category of danger and mandatory respectively.

Traffic Sign Dataset (¼ page)

The trained models in this research were evaluated using two datasets, the modified Belgium Traffic Sign Recognition Benchmark (BTSRB) dataset and the Australian Traffic Sign (ATS) dataset. The specifications for these two datasets are provided below.

Dataset	Size	Colour	Numbers of	Numbers of	Numbers of

			Image	Shape	Sign
BTSRB	28 * 28	Greyscale	7430	5	16
ATS	28 * 28	Greyscale	135	5	16

It is crucial to notice that the data provided in the BTSRB dataset experiences severe imbalance, which could affect the performance of the model being trained on this dataset. The lowest distributed sign-shape is *hex*, which only corresponds to nearly 2% of the whole dataset, while *round* corresponds to nearly 48%. The same imbalance is experienced with sign-type, where *warning* corresponds to nearly 19% of the dataset, while *crossing* only corresponds to 2.5%.

Experiment (1.5 page)

The experiment conducted in this paper is performed on Google Colab, with Intel(R) Xeon(R) CPU @ 2.20GHz. The experiment was set up to evaluate the performance of three based models, a multiple layer perceptrons neural network versus two popular convolutional neural network architectures which are the VGGnet architecture and a simplified LeNet architecture. The base architectures were trained using part of the BTSRB dataset (60% of the 80% training data) and modified depending on its performance on the validation set (20% of the 80% training data from BTSRB dataset) and a summary and the reason behind these modifications are presented in this section (please refer to Appendix A for detailed model performances).

Multiple Layer Perceptrons

The base model is a classic 3 layers MLP consisting of input, output layers and one hidden layer. The hidden layer has 32 perceptrons with sigmoid as an activation function whereas the output layer has softmax as an activation function. The model performs well on the training set with an accuracy of roughly 95%, however, only gets an accuracy of roughly 90% on the validation set. This means that this model experiences a slight overfit. In addition, the model only starts to converge after 100 epochs. The purposes of parameter tuning would be to find a balance between further improving the accuracy on the training set, as well as reducing overfit and improving convergent time.

Class weight, hidden layers and perceptrons

With the purpose of improving the accuracy of the training set, hidden layers and perceptrons were added to the base model. The numbers of hidden layers were increased to 2 and the numbers of perceptrons were increased to 128. In addition to adding layers and perceptrons, class weights were also considered in this tuned model since the training data experiences quite severe imbalance. Although the performance on the training set did improve as expected to nearly 100%, the model also takes longer to converge. This is the reason why *Relu* activation function was experimented, since *Relu* is not subjected to the problem of vanishing gradient despite the fact that the network is not very deep. Moreover the gap between the performance on the training set and validation set also increased. For this reason, different techniques to reduce overfitting were employed.

Relu

Results (1 page)

Ultimate Judgement (1 page)

Conclusion (½ page)

Reference Lists

[1] https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6707049

[2] https://ieeexplore-ieee-org.ezproxy.lib.rmit.edu.au/stamp/stamp.jsp?tp=&arnumber=7486576

D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber. A committee of neural networks for traffic sign classification. InProceedings of International Joint Conference on Neural Networks (IJCNN). 2011

D. Li, D. Zhao, Y. Chen and Q. Zhang, "DeepSign: Deep Learning based Traffic Sign Recognition," *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1-6, doi: 10.1109/IJCNN.2018.8489623.

Y. Wu, Y. Liu, J. Li, H. Liu and X. Hu, "Traffic sign detection based on convolutional neural networks," *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 2013, pp. 1-7, doi: 10.1109/IJCNN.2013.6706811.