Group 10

Traffic Sign Identification Using Deep Learning

Final Report

Myles Borthwick, Kenneth Loughery, Dylan Hofer, Gabriel Mathias, Nusrat Zerin Zenia

1. Introduction

With the advent of deep learning technology, automotive companies have started to shift their focus towards automated vehicles to improve road safety by eliminating operator error, the leading cause of vehicular accidents [1]. However to enable the automated vehicles to operate smoothly in the current landscape of transportation they must be able to follow the same traffic rules current drivers adhere to. While there are many areas of consideration in respect to automated vehicles, successfully detecting and recognising traffic signs can be considered a fundamental starting point.

1.1 Literature Review

Traffic signs were designed to be easily detected by human drivers, teaching a machine to do the same identification can be tricky at times due to the nuances that human brains can process automatically. Signs vary in color, shapes, text, and icons which leads to many different classes of signs. Sometimes, a subset of classes does not vary much in appearance as they convey similar yet different meanings (e.g. different speed limits). Furthermore, weather conditions and change of daylight make it more challenging to automatically differentiate traffic signs [2, 3].

To solve the aforementioned problems regarding the automatic detection and classification of traffic signs, several research efforts have been made. Convolutional Neural Networks (CNNs) have been proven to provide superior performance in traffic sign recognition [4, 5, 6]. For example, Mehta et al. [7] have proposed a CNN architecture for traffic sign classification with three convolutional layers each having 32 filters of size 3×3, and two fully connected layers. Convolutional layers were followed by max-pooling layers whereas a dropout layer is used between the fully connected layers. Belgium traffic sign dataset (BTSD) with 62 different classes was used to train and validate the model. The model accepts color images of size 64×64. Different optimizer and activation functions were used to evaluate the performance of the CNN model. The best result (97.06% accuracy) was achieved with adaptive moment estimation (Adam) optimizer and softmax activation function. Yang et al. [8] have applied a color probability model to reduce the processing time of traffic sign detection and classification. The color probability model [9] maps the color image to a grayscale image where the pixels of traffic signs are assigned high intensities but the background is assigned low intensities. From these gray images, Maximally Stable Extremal Regions (MSERs) were extracted as traffic sign proposals and a support vector machine was used to classify the detected signs into three superclasses based on the histogram of oriented gradients (HOG) feature. Moreover, the results of the SVM classifier were further applied to three separate CNN models for subclass classification. Each CNN model consists of two convolutional layers of 32 filters of size 5×5; followed by two subsampling layers and two fully connected Multilayer Perceptron (MLP) layers. The empirical experiment for classification was performed on the German Traffic Sign Recognition Baseline (GTSRB) dataset and the model achieved 97.75% accuracy with 3ms per image classification time which is much faster than other state-of-the-art models. Zhang et al. [10] have proposed a lightweight model consisting of five convolutional layers, each followed by max-pooling and batch normalization layer. The model achieved an accuracy of 99.61% and 99.13% in GTSRB and BTSD respectively. Wong et al. [11] have proposed Micronet, 16-bit floating-point microarchitecture with numerical and spectral optimization. The model consists of five convolutional layers where the first layer is a point-wise convolutional layer containing a kernel of size 1×1 followed by spatial convolutional layers with a kernel size of 5×5. Rectified Linear Unit (ReLU) was used as an activation function. The model achieved 98.9% accuracy on the GTSRB dataset. Besides, Aghdam et al. [12] have implemented a sliding window to reduce the computational time of CNN and make the classification faster. The proposed model consists of three convolutional layers and two fully connected layers and the model takes grayscale images as input. The first layer has a kernel size of 5×5 but the others are 3×3. Parametric ReLU (PReLU) was used for activation. Xu et al. [13] confirm that histogram equalization can balance the brightness and contrast of traffic sign images which contribute to enhancing classification accuracy. Alghmgham et al. [14] have developed a new dataset consisting of 2718 traffic signs of Saudi Arabia (SA-TRS-2018) and proposed a CNN model for classification. The model includes two convolutional layers, two max pooling layers followed by a dropout layer and three dense layers. The model achieved almost 100% accuracy on the new dataset, however the performance of the model needs to be verified on some benchmark datasets. Moreover, Alvaro et al. has achieved 99.71% accuracy on GTSRB. The proposed model includes three convolutional layers each followed by max pooling and local contrast normalization. However, this research domain still requires further empirical studies to innovate a benchmark deep learning architecture for traffic sign classification. Our aim is to develop a deep learning model to reach the human performance in detecting traffic signs. A summary of the related works is outlined in Table 1.

Table 1: Summary of the reviewed articles

Ref.	**Dataset**	**# CL**	**#FC**	**Optimizer**	**Dropout**	**Pooling**	**Activation**	**Accuracy**
[7]	BTSD	3	2	Adam	0.3	max	ReLU	97.06%
[8]	GTSRB	2	2	n/m	n/m	L-2	n/m	97.75%
[10]	GTSRB BTSD	5	1	Adam	n/m	max	n/m	99.61% 99.13%
[11]	GTSRB	5	1	SGD	n/m	n/m	ReLU	98.9%
[12]	GTSRB	3	2	n/m	n/m	max	PReLU	99.34%
[14]	SA-TRS-2018	2	3	RMSProp	0.2	max	ReLU	~100%
[15]	GTSRB	3	2	SGD	n/m	max	ReLU	99.71%

Note: #CL: number of convolutional layers; #FC: number of fully connected layers; n/m: not mentioned

2. Material and Methods

This section describes the dataset, experimental setup and the proposed CNN architecture for traffic sign classification.

2.1 Dataset

In this project we are using the GTSRB dataset [2]. This dataset has been widely used by many research efforts and regarded as a benchmark dataset for traffic sign classification [14-16]. A sample of the images contained in the dataset is shown in Figure 1. The dataset of the traffic signs are classified into 43 different classes. The distribution of the 43 classes can be seen in Figure 2. The dataset itself comes with a development (75%) and test set (25%) already split. A detailed specification of the dataset is given below:

- 51,840 labelled images
- 43 classes
- Image size is between 15 x 15 and 250 x 250 pixels
- Traffic signs may not be centered in the image
- Annotations and images are separated; annotations are done in a csv that corresponds to the image file, and includes image size information

Figure 1: Sample images of GTSRB dataset.

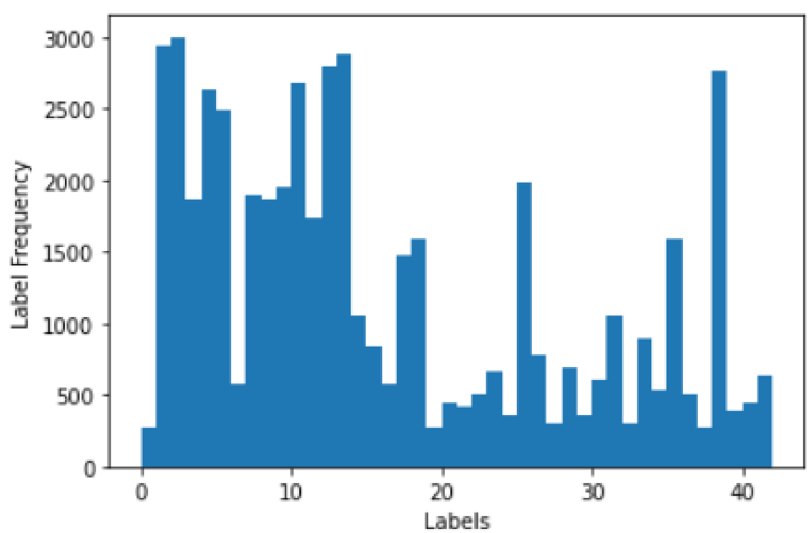

Figure 2: Class distribution of GTSRB dataset.

2.2 Experimental Setup

The libraries imported to aid in the creation of the machine learning model include Keras and TensorFlow. The dataset images were resized to a standard 48x48 pixels with the RGB color channels. Following resizing, min-max scaling was performed for data normalization. The dataset had already been shuffled and randomly split into a 25% test set and a 75% development set. The development set was further split into a 25% validation set and a 75% training set.

For data augmentation, it should be emphasized that the information presented by traffic signs will be affected by their orientation; therefore it is imperative that the augmentations applied to the images do not go so far as to change the information provided by the sign itself. With this in mind, the image adjustments like flipping, rotation, magnification, and shifting, which were considered originally, needed to be re-evaluated. Horizontal or vertical flipping augmentations would make several signs lose all meaning, so those augmentations were no longer in consideration. Rotational adjustments were also applied, but the angle of rotation was limited to 11 degrees so as to not to fully distort the integrity of the traffic sign. Beyond that, mild shearing, shifting, brightening, darkening, and magnification of 10% was applied to the dataset as well. Another augmentation investigated was zero component analysis (ZCA) whitening as it is an often useful feature or edge extraction tool, however it was found that this augmentation yields quite poor results for the dataset. Figure 3 displays some samples from the augmented data. The categorical values were represented utilizing one hot encoding. Due to the large amount of data to be processed, the computation time to train the model was extremely large. By using the GPUs provided in GoogleColab the team decreased training time and was able to test different iterations of the model at a faster rate.

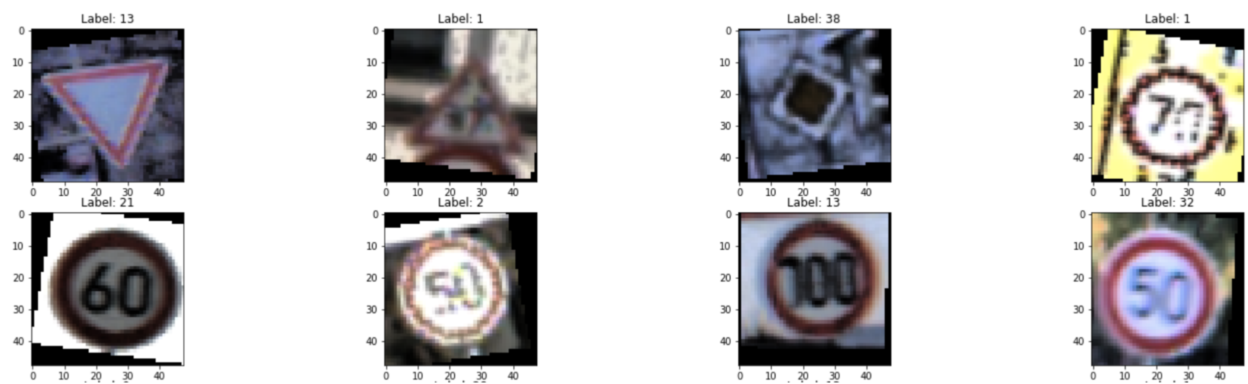

Figure 3: Sample images of GTSRB augmented dataset.

2.3 CNN Architecture

There are a multitude of architecture modifications that are possible with the CNN model. Within the literature, the best model that has ever been designed with the dataset is a CNN with 3 spatial transformers [15]. This model outperformed even the human benchmarks (98.81%) that were provided [3], one of only two models able to achieve this feat. The other, which was implemented during the competition this dataset was created for, used a multi-column CNN, and scored the highest within the competition itself [16]. These models offer encouragement to our choice of the CNN model. Our model, which is displayed in Figure 4, consists of six convolutional layers and two fully connected layers. Each convolution layer had its kernels initialized with the He uniform variance scaling initializer, and each layer had a keras L2 regularization penalty applied to it. Each of the layers utilized ReLU activation, and were succeeded by a batch normalization, dropout, and max pooling layer; we used an increasing dropout coefficient that increased towards the end of the model architecture. This, as well as the L2 regularization, was used to reduce overfitting.

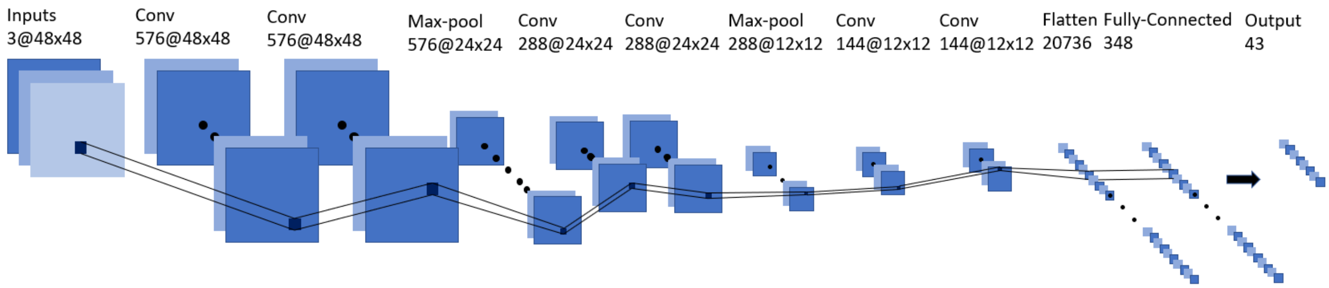

Figure 4: Representation of the model architecture tested

2.4 Model Training

The proposed CNN architecture was trained for ~150 epochs with a batch size of 32 which is the point where the callbacks were activated as the model stopped improving its accuracy. Stochastic gradient descent with a high momentum optimizer was applied to train the model. The base learning rate was set to 1e^-4 and a scheduled decay halved this rate every ten epochs. The momentum was set to 0.9.

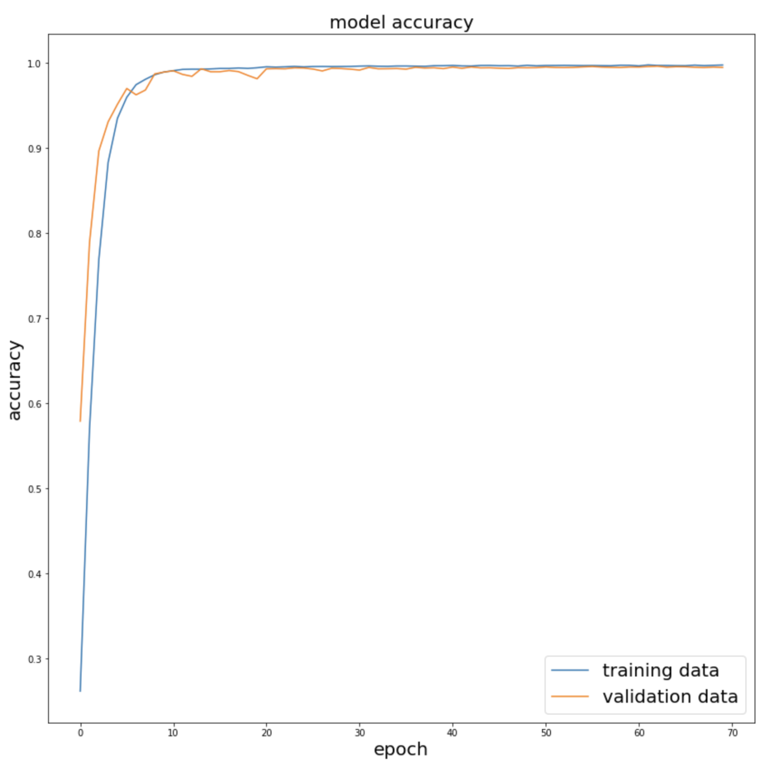

Figure 5: Graph of the model's accuracy scores for the training and validation data over the first 70 running epochs.

3. Results and Discussion

This section details and discusses the results and outlines the future considerations for this project.

3.1 Results

The CNN model described in Figure 4 was able to achieve excellent accuracy values for the test dataset, with the highest accuracy being 97.97% (found after the final presentation was filmed, prior it was 97.7%). Figure 5 shows the model's accuracy for the training and validation data over the first 70 epochs, which is increasing steadily, showing that the model is not overfitting. To get to our final model, weeks of iterations were conducted to find the best possible score, using different model sizes, different dropout ranges, and different regularization parameters. In our final model, our dropout coefficient increased by 0.1 with every layer, starting at 0.2 and finishing at 0.5 for the fully connected 348 kernel layer. Likewise, our l2 regularization increments by 0.1 from 0.1 to 0.4 for our fully connected layer.

To continue to improve the score, numerous data augmentation tweaks were also attempted, ranging from different shear, zoom, width, height, and fill adjustments, and also involving a few different models. Though the data augmentation ultimately provided no further improvements, achieving a maximum accuracy of only 97.5%, this process showed that our model was as fitted as possible to the exact format of the GTSRB dataset (for better or worse). When our current model is compared to the model presented in the midterm report, the current model incremented its training and validation accuracies as well as the testing accuracy, moving from a training accuracy of 92.6% before, to over 99% now, and from 97.6% to 99.6%. This showed a good development in our overall fitting to the data, and helped provide the 3.7% improvements seen in the final model accuracy (from 94.3% to 97.97%).

Git repository for final notebook: https://github.com/kenloughery/ENEL645_FinalProject.git

3.2 Future Considerations

The current model described in this report serves as a proof of concept for a small slice of the machine learning necessary for developing functional and safe automated vehicles. Other relevant machine learning problems for automated vehicles may include the following:

- **Object Detection** - This is a vital component for automated vehicles as it is a driving mechanism for environmental awareness.
- **Object Localization and Movement Prediction** - Automated vehicles must be able to understand movement and the relative positions of objects in order to avoid collisions.

- **Driver Monitoring -** Facial recognition models could be used for both vehicle theft prevention as well as identifying driver impairment or injury.
- **Predictive Failure Analysis** - Complex systems may also involve models used to predict potential failure points within the vehicle before they pose a serious risk. Although this may not be completely necessary for the first adoption of self-driving vehicles, it would still prove useful in improving the overall safety of these machines.

Full development and integration of automated vehicles into current road infrastructure is an extremely complex process. Even once all factors have been addressed through different models, each one needs rigorous testing and modification for it to achieve an acceptable standard and be deemed safe for the road.

4. Conclusion

The final model developed through this project achieved a 97.97% test accuracy. This improves upon the preliminary model developed in the first phase of this project by 3.7%, which has exceeded the targeted increase of 2%. In regards to the 98.84% average human benchmark [3], the final model only slightly underperforms suggesting that further development of this model or subsequent models could lead to results that exceed the ability of human drivers. While this model only fulfills one aspect of the machine learning technology needed for safe automated vehicles, it successfully proves the viability of machine learning for following the traffic rules in the current landscape of transportation.

References

[1] Ryan, M. The Future of Transportation: Ethical, Legal, Social and Economic Impacts of Self-driving Vehicles in the Year 2025. Sci Eng Ethics 26, 1185–1208 (2020). https://doi.org/10.1007/s11948-019-00130-2

[2] C. Liu, F. Chang, Z. Chen and D. Liu, "Fast Traffic Sign Recognition via High-Contrast Region Extraction and Extended Sparse Representation," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 79-92, Jan. 2016, doi: 10.1109/TITS.2015.2459594.

[3] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Netw.*, vol. 32, pp. 323–332, Aug. 2012, doi: 10.1016/j.neunet.2012.02.016.

[4] Safat B. Wali, Mahammad A. Hannan, Aini Hussain, Salina A. Samad, "An Automatic Traffic Sign Detection and Recognition System Based on Colour Segmentation, Shape Matching, and SVM", *Mathematical Problems in Engineering*, vol. 2015, Article ID 250461, 11 pages, 2015. https://doi.org/10.1155/2015/250461

[5] D. Zang, Z. Wei, M. Bao, J. Cheng, D. Zhang, K. Tang, and X. Li, "Deep learning-based traffic sign recognition for unmanned autonomous vehicles," Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, vol. 232, no. 5, pp. 497–505, 2018.

[6] A. Shustanov, P. Yakimov, CNN Design for Real-Time Traffic Sign Recognition, in Procedia Engineering, vol. 201, pp. 718-725, 2017, doi: 10.1016/j.proeng.2017.09.594.

[7] S. Mehta, C. Paunwala and B. Vaidya, "CNN based Traffic Sign Classification using Adam Optimizer," *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, Madurai, India, 2019, pp. 1293-1298, doi: 10.1109/ICCS45141.2019.9065537.

[8] Y. Yang, H. Luo, H. Xu and F. Wu, "Towards Real-Time Traffic Sign Detection and Classification," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 2022-2031, July 2016, doi: 10.1109/TITS.2015.2482461.

[9] Y. Yang and F. Wu, "Real-time traffic sign detection via color probability model and integral channel features," in Pattern Recognition, vol. 484, ser. Communications in Computer and Information Science. New York, NY, USA: Springer-Verlag, 2014, pp. 545–554.

[10] Zhang, J., Wang, W., Lu, C. et al. Lightweight deep network for traffic sign classification. Ann. Telecommun. 75, 369–379 (2020). https://doi.org/10.1007/s12243-019-00731-9

[11] A. Wong, M. J. Shafiee and M. St. Jules, "MicronNet: A Highly Compact Deep Convolutional Neural Network Architecture for Real-Time Embedded Traffic Sign Classification," in *IEEE Access*, vol. 6, pp. 59803-59810, 2018, doi: 10.1109/ACCESS.2018.2873948.

[12] H. H. Aghdam et al., A practical approach for detection and classification of traffic signs using Convolutional Neural Networks, Robotics and Autonomous Systems, vol. 84, pp. 97-112, 2016, doi: 10.1016/j.robot.2016.07.003.

[13] Xu, H., Srivastava, G. Automatic recognition algorithm of traffic signs based on convolution neural network. *Multimed Tools Appl* 79, 11551–11565 (2020). doi: 10.1007/s11042-019-08239-z

[14] D. A. Alghmgham et al., Autonomous Traffic Sign (ATSR) Detection and Recognition using Deep CNN, in Procedia Computer Science, vol. 163, pp. 266-274, 2019, doi: 10.1016/j.procs.2019.12.108.

[15] Á. Arcos-García, J. A. Álvarez-García, and L. M. Soria-Morillo, "Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods," *Neural Netw.*, vol. 99, pp. 158–165, Mar. 2018, doi: 10.1016/j.neunet.2018.01.005.

[16] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Netw.*, vol. 32, pp. 333–338, Aug. 2012, doi: 10.1016/j.neunet.2012.02.023.

[17] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale Convolutional Networks," in *The 2011 International Joint Conference on Neural Networks*, San Jose, CA, USA, Jul. 2011, pp. 2809–2813, doi: 10.1109/IJCNN.2011.6033589.

Contributions

Myles Borthwick

- Participated in topic research
- Report Writing (Introduction, discussion, experimental setup, future considerations etc.)
- Helped with development and testing of models
- Edited and formatted the report

Consensus Score: 3

Ken Loughery

- Helped with research
- Contributed to model development for google colab
- Refined model & augmentation parameters to improve output
- Editing of the report

Consensus Score: 3

Dylan Hofer

- Participation in topic research
- Data organization/access
- Assisted with data augmentation of the model
- Editing and formatting of report

Consensus Score: 3

Gabriel Mathias

- Helped with researching the topic
- Built a working CNN model with appropriate preprocessing
- Data augmentation section
- Graphic of the model architecture
- Editing of the document

Consensus Score: 3

Nusrat Zerin Zenia

- Participated in topic research
- Literature review and appropriate citation
- Helped with development of the model
- Report writing

Consensus Score: 3