**The University of Memphis**

**COMP 7/8740 –Neural Nets**

**Project Report: Traffic signal classification for autonomous driving**

**Talha Chaudhry, Applied Statistics, Fall 2020**

**Project Goal and Scope**

The goal was to train a neural model to classify single RGB images of traffic signs into 43 classes. This was a classification problem, that is, given an image, the network should classify the traffic sign as a sub-task for autonomous driving systems. To achieve this a Convolutional Neural Network (CNN) was trained on extracted features from a training data set of images, which was 76% of the entire dataset. The CNN’s performance was then assessed on extracted features of the remaining 26% dataset, the assessment metric was ‘accuracy’.

**Literature Review**

****

**Fig 1: Images from the GTSRB Dataset**

For this project, the German Traffic Sign Recognition Dataset (GTSRB) was used which is available [here](http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset) [1]. The images are of different sizes and clarity. Some images were taken in the dark and others are quite fuzzy (see Figure 1). Another characteristic of the dataset was that for the 43-different traffic signs the number of images varied. Total number of images were 51,839 with 39,209 images for training and 12,630 for testing.

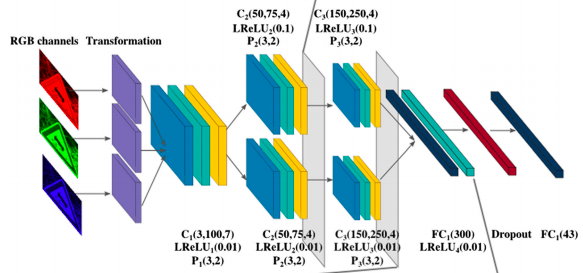
Due to the varying sizes of images they were at first resized to 32X32X3. There after features were extracted, that is, the pixel values were obtained for each pixel. Then ‘[Histogram Equalization](https://en.wikipedia.org/wiki/Adaptive_histogram_equalization)’ was performed, an equalization technique that greatly improves contrast in images [2]. This method computes histograms each of which corresponds to a different section of the image and then redistributes the lightness of the image [2]. The final step for the extraction is normalization of these post-equalization pixel values. Normalization ensures that the pixel values are from the same distribution and it makes computations less costly especially reducing the number of multiplications [5].

The 43 classes, which are numbered 0 to 42, are converted to a 43 character long using “One-Hot Encoding” where one of the characters is 1 corresponding to its class while the rest are 0.

Foucher et all (2010) had proposed a multiple-algorithm solution to a similar problem: Road Sign detection [3]. The road signs considered were: crosswalks, and arrows on the road. They define three aspects of the image that distinguish a road sign by shape, color, and text. The following algorithms are used: Contour Fitting, Radial Symmetry Transform, and pair-wise voting scheme. The authors report an accuracy of 95% on crosswalks and 87% on arrows. Bruno, Diego & Osorio, Fernando (2017) used an Adaptive Deep Learning System which was combination of CNNs and LSTMs (Long Short-Term Memory) [4]. They used the GTSRB dataset and reported an accuracy of 97.24%.

**Case Study**

Aghdam, Hamed & Heravi, Elnaz & Puig, Domenec. (2016) suggest multiple CNNs and ‘ensemble’ them [5]. In their proposed solution, they use a ‘Leaky ReLU’ transfer function. This transfer function uses modular arithmetic and Taylor’s series expansion up to a specified degree of polynomial. This function allows for some error to be tolerated. The authors show that this allows for greatly reducing arithmetic operations including multiplications. They are also proposed in their solution a specific architecture on the CNNs (See Fig 2).



**Fig 2:** Architecture of CNN. Once the images are transformed and resized

and features extracted they are passed through multiple combination of

layers. Each combination consists of convolutional layer, an activation layer,

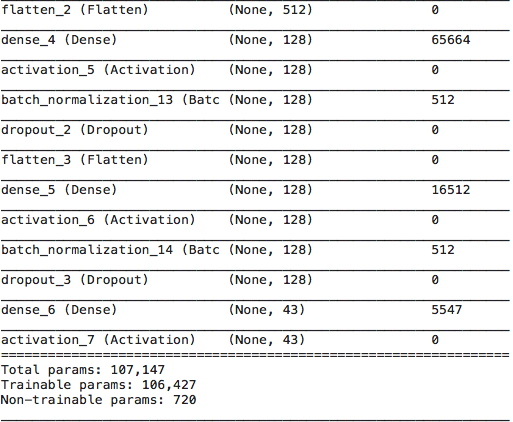
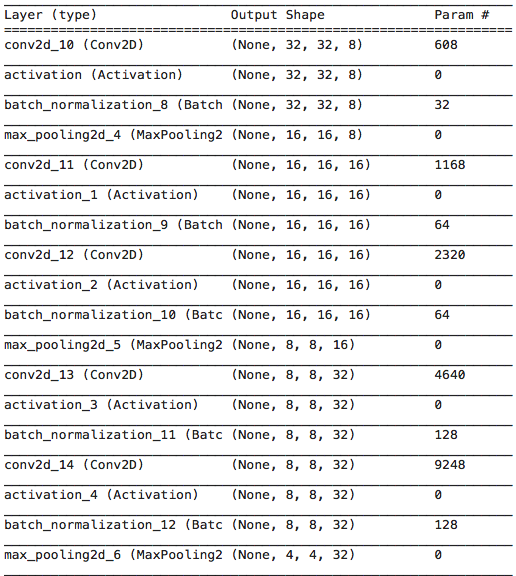
and a pooling layer [5].

Figure 2 shows the proposed architecture; the output is a ‘coarse output’. The total number of neurons is 43 corresponding to each class. When an image passes through the CNN only one output neuron fires which can be interpreted as 1 while the rest of the 42 are considered as 0. This string corresponds to one of the 43 classes. The gray layers represent normalization. Notice that there is a splitting in the layers, this is one aspect of the ensemble as that corresponds to another CNN.

The authors state that their accuracy on the testing set, with 80% reduced arithmetic operations, as 99.07%. With 95% reduced arithmetic operations, the accuracy achieved was approximately 98.6%. At the time of publishing this was the state-of-the-art solution.

**Take- Home Deliverable**

Following on from Aghdam et all (2016) a similar architecture was used for building a CNN as a solution for the goal, that is, a network that would classify traffic signs [5]. 76% of the GTRSB data was used for training and the remaining 24% was used for testing the quality of the solution. Since this is a single CNN, there was no ensembling. The architecture consists of the input layer followed by a convolutional layer then a ReLU activation layer and ending with Normalization. After this two sets of layers, where each set consists of a convolutional layer, then a ReLU activation layer and Normalization. Followed by the same combination. After each set a pooling layer is added. At the end flattening of the layer to get a ‘dense’ (fully connected) layer. This flattening is done twice, and finally the output layer which is a ‘coarse output’ signifying the 32 classes. See Figure 3 for more detail.



Aghdam et all (2016) refer to several papers that have used similar CNN based architectures as solution to this classification problem of the GTSRB dataset [5]. They list accuracy metrics of these papers as ranging from around 91% to approximately 99%. Therefore, it was considered acceptable if at least a 92% accuracy was achieved on the testing dataset. In the following table, the accuracy measures for 20 and 30 epochs are provided.

|  |  |
| --- | --- |
| Epochs | Accuracy |
| 20 | 94.61% |
| 30 | 95.03% |

Given the accuracy achieved, the solution is considered acceptable.

**TIMELINE Date (completed)**

Project Proposal Oct 02

Lit Review completed Oct 17

Case Study Oct 25

Final Project Oral Nov 24

Final Report uploaded to elearn Nov 24, 2020.

**REFERENCES**

**[1]** [J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, pages 1453–1460. 2011](http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.464.2017).

**[2]** <https://en.wikipedia.org/wiki/Adaptive_histogram_equalization>

**[3]** [Foucher, Philippe & Tarel, Jean-Philippe & Soheilian, Bahman & Charbonnier, Pierre & Paparoditis, Nicolas & Belaroussi, Rachid. (2010). Road Sign Detection in Images : A Case Study. ICPR'10. 10.1109/ICPR.2010.1125](https://www.researchgate.net/publication/48418194_Road_Sign_Detection_in_Images_A_Case_Study)**.**

**[4]** [Bruno, Diego & Osorio, Fernando. (2017). Image classification system based on deep learning applied to the recognition of traffic signs for intelligent robotic vehicle navigation purposes. 1-6. 10.1109/SBR-LARS-R.2017.8215287](https://ieeexplore.ieee.org/document/8215287).

**[5]** [Aghdam, Hamed & Heravi, Elnaz & Puig, Domenec. (2016). A Practical and Highly Optimized Convolutional Neural Network for Classifying Traffic Signs in Real-Time. International Journal of Computer Vision. 122. 10.1007/s11263-016-0955-9.](https://link.springer.com/article/10.1007/s11263-016-0955-9)