ISSN:0975-9646

Traffic Sign Detection Using Multi-Layered Convoluted Neural Network

Aryan Arora

Abstract: Traffic sign detection and recognition is key towards the development of full self-driving for advanced driver assistance systems (ADAS). In this paper, we have developed a multi-layer convoluted neural network to train on 4170 images of traffic signal detection and recognition with an accuracy of upto 98%. To enable this, we have developed multi-layer convoluted layers with four convoluted blocks and classified the images into 9 labels. We trained our model on these images and the multi-layered CNN network was able to classify the images with a very high accuracy.

Nomenclature:
ANN – Artificial Neural Network
MLP – Multi Layer Perceptron
DTI – Debt-to-income ratio
SVC – Support Vector Classifier
LR – Logistic Regression
KNN – K-Nearest Neighbor
ROC – Receiver Operating Characteristics
AUC – Area Under Curve
LR – Logistic Regression

Keywords: Prediction, Classification, Feature Engineering, Machine Learning, Deep Learning, ANN, SVM, microfinance, random forest, logistic regression.

1. INTRODUCTION

With the advent of the 2010s, technology became heavily on automation. The key motivations behind automation include improved efficiency, reducing redundancies, cost-saving, etc. In this regard, computer vision has a large number of applications in webcams, security cameras, drones, mobile phones, cars, etc. Full self-driving (FSD) [1, 2, 3, 4, 5] is looked upon as the next innovation in automobiles. There are 5 automation levels in automobiles. Level 5 self-driving will enable driverless cars while guaranteeing safe travel [6, 7]. There are multiple benefits of FSD, such as improved fuel consumption, higher mileage, safer roads for both passengers and pedestrians, etc. This is achieved with the help of computer vision and artifical intelligence. First, cameras placed around the car gather information about the car's surroundings. Then, the artificial intelligence chip within the car constructs a 3D map[9] of its surrounding which enables the car to navigate in the real world without crashing. To ensure self-driving cars follow traffic rules, it is critical that computer vision recognise traffic signs [8, 9, 10] and accordingly make decisions to ensure compliance. In this paper, we will detect and classify Chinese traffic signs using a multi-layered convoluted neural network (ML-CNN).

Lately, to enable object detection, R-FCN [11], YOLO [14], SSD [13], R-CNN [12] employed convoluted neural networks (CNN) in mobile devices, drones and others as object detection cameras. As the images occupy much large memory space compared to non-pictorial data, the computational cost and time both increase significantly. Therefore, efficiency and accuracy both are key when implementing an image classification or object detection algorithm. Krizhevsky et. al., published an article in 2012 on ImageNet dataset classification which is popularly known as AlexNet which was trained on 60 million parameters and had an error rate of 15.3%. Later, in 2014, Zeiler and Fergus [16] demonstrated convoluted network visualisation (which is popularly known as ZF Net), which beat the state-of-the-art results trained on Caltech-101 and Caltech-256 datasets. They demonstrated deconvolutional technique for visualisation of CNN layers. K. Simonyan & A. Zisserman [17] demonstrated a very deep convoluted network for large-scale image recognition using small convoluted filters of 3x3 compared to AlexNet's 11x11 filter. The depth of the layer was 16-19 weighted layers because of which it is popularly known as VGG16 (for 16 layers) and VGG19 (for 19 layers). This laid the foundation for GoogleNet and the concept of "inception modules" introduced by Szegedy et. al., in 2015 [18]. Instead of stacking convoluted layers with pooling it introduced an inception module which was a innovative way of creating novel CNN architecture. This added complexity to filter selection and usage for convoluted layers. It consisted of 22 convoluted layers.

Finally, there is the popular ResNet structure demonstrated by K. He et. al., from Microsoft's research in 2015. The ResNet consists of 152 weighted layers with decreased complexity and an improved
error rate.

2. MULTI-LAYERED CONVOLUTED NEURAL NETWORK

Convoluted neural networks (CNN) belong to the class of deep learning techniques. CNN is mostly used for image recognition, pattern recognition and object detection. It is a classification technique to classify any given trained pattern into well-known labels. CNN is often employed for spatial data arranged in a grid like topology [20, 21]. In case of an image or photo, spatial data is a pixel having specific red-blue-green (RGB) values which are arranged in a grid format with different resolutions. The regular arrangement of these pixels appears like single image to human vision. A fundamental CNN architecture consists of an input layer, convoluted layer, pooling layer, flatten layer, fully connected layer and an output layer. An overall graphical representation is described in the figure 1.

Each layer is further described in detail in the next sub-sections below.

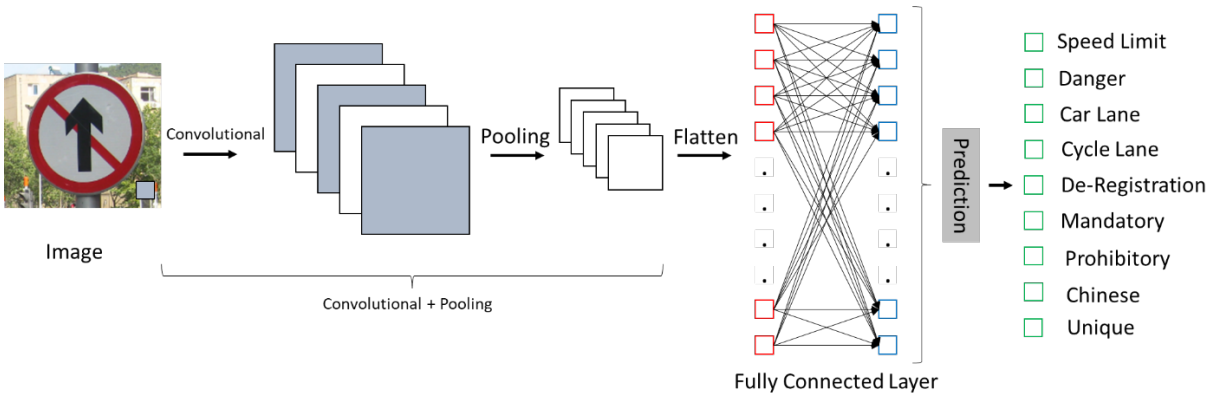


Figure 1: Implementation of basic CNN layer on image classification and recognition.

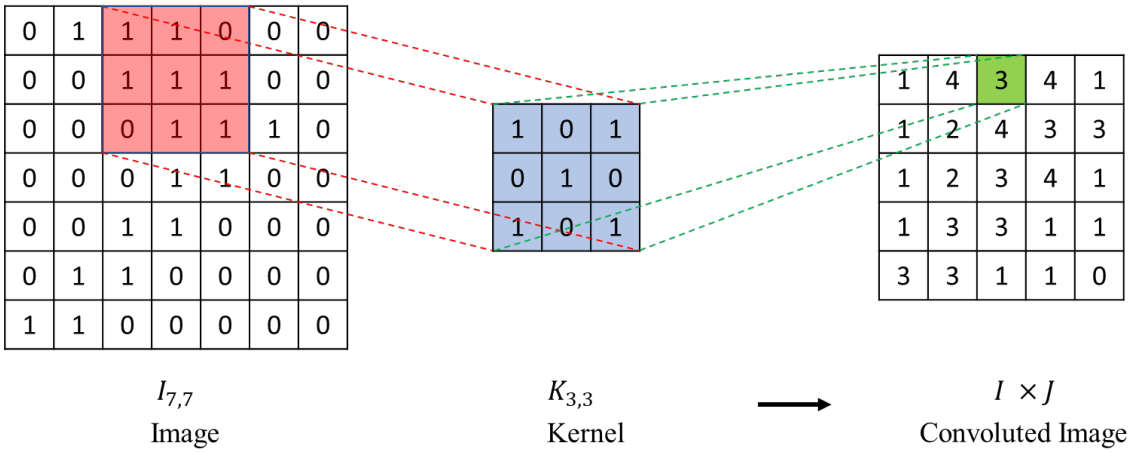


Figure 2: The convoluted image (I x J) produced after applying the kernel K to original image I.

2.1. Convolutional Layer

The convolutional layer is also known as the feature extraction layer from the input image. A filter of 2x2 or 3x3 matrix with some pre-defined weights is used to extract features. The weight value in the filter need not be same as the size of the filter. It can change from one input image to another depending upon what key feature needs to extracted. The filter is also known as the kernel parameter in the convolutional layer of the neural network. For a two-dimensional image,

I of resolution u, v, the kernel of size a x b, the convoluted image I x K is computed to be e, the summation of element-wise multiplication between the image and the kernel. This can be represented by equation 1,

$$I_{u,v} \times K_{a,b} = \sum_{i=1}^{u} \sum_{j=1}^{v} (K_{ij} \times I_{x+i-1,\ y+j-1}) \quad (1)$$

The convoluted image $I_{u,v} \times K_{a,b}$, is computed by overrunning the kernel over the input image after specifying stride length. Also, to enable non-linearity in the convoluted image rectified linear function is used which is commonly known as ReLU function. As the kernel runs across the image the weights remain constant. The convoluted image is transformed because of the application of the weighted kernel. Reduction in the number of parameters in the model also reduces complexity and error rate. With the application of ReLU function, after passing the kernel function the CNN network can easily detect the boundaries, lines and other distinctive features which is key to object detection and recognition.

2.2. Pooling Layer

The pooling layer is used to further down sample the images spatially. The pooling function leads to reduction in number of parameters that will be fed into the neural network which decreases complexity, thereby enabling feature detection which is immune to scale and orientation of the image. In simple terms, it generalizes low-level information that can be successfully used in a neural network. Similar to convolution, in pooling we use a filter matrix on the already convoluted image. Instead of summation of the elementwise product we use the concept of max-pooling, average pooling or sum pooling. Max-pooling is the most common pooling technique used, which is also used in this paper.

A filter of 2x2 matrix is moved over the input image with a stride of (1,1). This means that the filter will move by one-step in x or y-direction over the input image. Stride of (2,2) means the filter will move over the image by 2-step to prevent overlapping. Having larger strides will lead to smaller output image. In max-pooling the maximum value across all matrices is used as the output value. In average-pooling the average of all values is the output.

2.3. Flatten Layer

The flattening layer flattens the 2-D or 3-D matrix image into 1-dimension dataset which can also be called an *n x 1* tensor. This 1-dimension data acts as the input and is fed into the multi-layered neural network.

2.4. Fully Connected Layer

The 1-dimensional tensor output from the flattening layer is used as an input to the artificial neural network (ANN). The key purpose of the ANN is to combine all the features in the form of attributes so that the ANN can classify the images with higher accuracy for better prediction. The back-propagation and error are calculated and the feature detectors are optimized to improve the performance of the model. The total number of output parameters in our case is 9 as we have 9 labels (see figure 3).

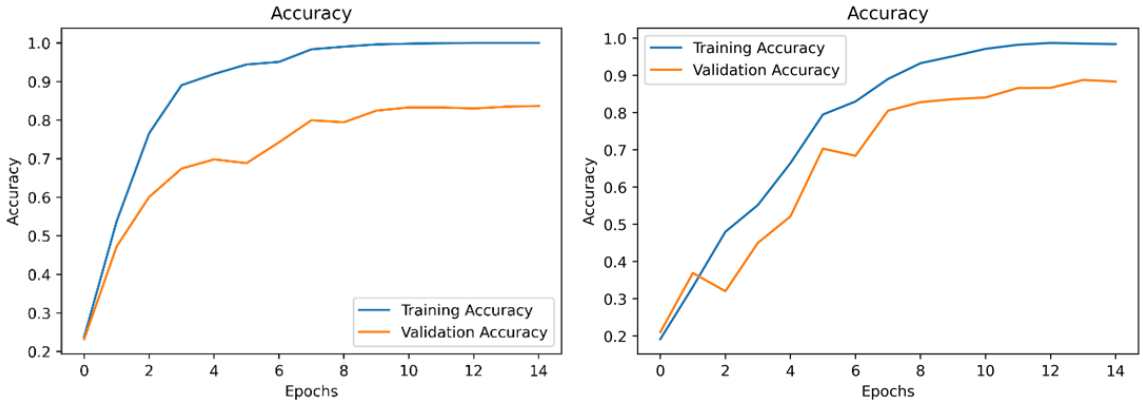
3. Experimental Methodology

Images are classified into 9 different folders. The folder label is used as a classification label in the training architecture following which the images are fed into the convoluted layers. In our model architecture there are 4 convoluted blocks and 1 neural network block along with an input and output layer.

3.1. Dataset

The Chinese traffic sign dataset is collected from website of National Nature Science Foundation of China (NSFC) website (link: Traffic Sign Recogntion Database (ia.ac.cn)). The dataset contains a total of 58 sign and includes total of 6164 traffic sign images. The dataset is further divided into training and testing sets. The training set contains 4170 images and the testing set contains 1994 images. For our modelling purpose we have further classified them into nine different labels which are labelled as speed limit, mandatory, prohibitory, danger, cycle lane, car lane, Chinese signs, derestriction and unique (see figure 3).

Figure 3: The signs are classified into 9 categories which are assigned in the architecture in the form of labels.



Dropout = 0.20 Dropout = 0.40

Figure 4: The accuracy of the multi-layered convoluted neural network with subsequent epochs. With the increased dropout rate the model was better optimized compared to lower dropout rate. It can be seen that although with training data the accuracy reaches ~98%, there are some discrepancies on the validation dataset.

3.2. CNN Architecture

The ANN architecture we build contains 4 convoluted block which includes a convoluted layer as well as a pooling layer. After every instance of pooling we use batch normalisation to ensure the output data is normalised. The neural network contains 100 input units with a dropout rate of 0.20. The dropout rate of 0.20 signifies that the 20% of the output is dropped before inputting it into the next neural layer. When going from 3 convoluted blocks to 4 convoluted blocks the total parameters dropped from 419,359 to 138,009. This decreases the complexity of the model and increases the accuracy which is evident and discussed in the result section.

3.3. Parameter Tuning

A Rectifier Linear function is used as the activation function in the convoluted layer with a kernel size of 5x5 with strides of (1,1). The pooling layer in the 1st and 2nd convoluted block has 'same' padding while for the 3rd and 4th block, we use 'valid' padding. For the third and fourth convoluted block we use a kernel of size 3x3 instead of 5x5. As we are dealing with a categorical classification dataset therefore, 'softmax' is used as the activation function for the output layer. The loss function is 'categorical_crossentropy' and the optimizer is 'adam'. The 'adam' optimizer stands for adaptive moment estimation and is used for the optimization of the gradient descent to reach the minima in an efficient manner. For a binary classification problem, the loss function would preferably be 'binary_crossentropy'.

4. Results And Discussion

We successfully developed and demonstrated the multi-layer convoluted neural network (ML-CNN) on a core i7 mobile processor with 16 GB of memory and 4 GB GPU memory. With a dropout rate of 0.20 the model appeared to be overfitting the training data, giving a low accuracy on validation data. Therefore, the dropout was increased to 0.40 resulting in better optimisation and accuracy.

For a dropout rate of 0.20, the overall simulation took 2280.92 s for 15 epochs with 1.32 million parameters. After increasing the dropout rate to 0.40 with 15 epochs the number of parameters decreased to 79,459 and the training time also decreased to 2009.75 s.

5. Conclusions

In conclusion, in this article we have developed and demonstrated a four layered convoluted neural network to enable traffic sign recognition and detection. With our ML-CNN model we have been able to achieve an accuracy of 98% on the training dataset and 89% on the validation dataset. We can use the ground work done to further improve and train the model on a larger dataset. This would increase the accuracy and decrease the overall error rate. We have also decreased the complexity of the model with lower parameter set even when the number of layers is significantly higher than a vanilla CNN architecture.

References

1. Eichberger, A.; Wallner, D. Review of recent patents in integrated vehicle safety, advanced driver assistance systems and intelligent transportation systems. Recent Pat. Mech. Eng. 2010, 3, 32–44.
2. Campbell, S.; Naeem, W.; Irwin, G.W. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. Annu. Rev. Control 2012, 36, 267–283. [CrossRef]
3. Olaverri-Monreal, C. Road safety: Human factors aspects of intelligent vehicle technologies. In Proceedings of the 6th International Conference on Smart Cities and Green ICT Systems, SMARTGREENS 2017 and 3rd International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS), Porto, Portugal, 22–24 April 2017; pp. 318–332.
4. Luo, Y.; Gao, Y.; You, Z.D. Overview research of influence of in-vehicle intelligent terminals on drivers' distraction and driving safety. In Proceedings of the 17th COTA International Conference of Transportation Professionals: Transportation Reform and Change-Equity, Inclusiveness, Sharing, and Innovation (CICTP), Shanghai, China, 7–9 July 2017; pp. 4197–4205.
5. Andreev, S.; Petrov, V.; Huang, K.; Lema, M.A.; Dohler, M. Dense moving fog for intelligent IoT: Key challenges and opportunities. IEEE Commun. Mag. 2019, 57, 34–41. [CrossRef]
6. Yang, J.; Coughlin, J.F. In-vehicle technology for self-driving cars: Advantages and challenges for aging drivers. Int. J. Automot. Technol. 2014, 15, 333–340. [CrossRef]

7. Yoshida, H.; Omae, M.; Wada, T. Toward next active safety technology of intelligent vehicle. J. Robot. Mechatron. 2015, 27, 610–616. [CrossRef]
8. A. De La Escalera, L. E. Moreno, M. A. Salichs, J. M. Armingol, "Road traffic sign detection and classification", IEEE Transactions on Industrial Electronics 44 (6) (1997) 848–859. doi:10.1109/41. 649946.
9. L. Zhou, Z. Deng, "Lidar and vision-based real-time traffic sign detection and recognition algorithm for intelligent vehicle", in: Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on, IEEE, 2014, pp. 578–583.
10. A. Arcos-Garcia, M. Soilan, J. A. Alvarez-Garcia, B. Riveiro, "Exploiting synergies of mobile mapping sensors and deep learning for traffic sign recognition systems", Expert Systems with Applications 89 (2017) 286–295.
11. J. Dai, Y. Li, K. He, J. Sun, R-fcn: Object detection via region-based fully convolutional networks, in: Advances in neural information processing systems, 2016, pp. 379–387.
12. S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in neural information processing systems, 2015, pp. 91–99.
13. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, A. C. Berg, SSD: Single shot multibox detector, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 9905 LNCS, 2016, pp. 21–37. arXiv:1512.02325, doi:10.1007/978-3-319-46448-0_2.
14. J. Redmon, A. Farhadi, Yolo9000: better, faster, stronger, arXiv preprint 1612.
15. Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, editors. Advances in Neural Information Processing Systems 25. Curran Associates, Inc.; 2012. p. 1097–1105. Available from: http://papers:nips:cc/paper/4824-imagenet-classificationwith-deep-convolutional-neuralnetworks.pdf
16. Zeiler M.D., Fergus R. (2014) Visualizing and Understanding Convolutional Networks. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8689. Springer, Cham. https://doi.org/10.1007/978-3-319-10590-1_53
17. L. Simonyan, A. Zisserman, "Very deep convolutional networks for large scale image recognition", 1409.1556, arXiv. https://arxiv.org/abs/1409.1556v6 .
18. C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594
19. K. He et. al., "Deep Residual learning for Image Recognition", 1512.03385, 2015, arXiv. https://arxiv.org/abs/1512.03385
20. Sermanet P, LeCun Y. Traffic sign recognition with multi-scale Convolutional Networks. In: The 2011 International Joint Conference on Neural Networks; 2011. p. 2809–2813.
21. Lecun Y, Jackel L, Cortes C, Denker J, Drucker H, Guyon I, et al. Learning Algorithms For Classification: A Comparison On Handwritten Digit Recognition. 2000 07.