Proceedings of the 2nd International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/76/20240617

A traffic and road signal recognition method through deep learning

Wei Jiang

Aerospace Engineering, Beijing Institute of Technology, Beijing, China

1120212953@bit.edu.cn

Abstract. Before deep learning became popular, some researchers used traditional computer vision algorithms, including support vector machines, decision trees, and random forests to deal with traffic and road signal recognition problems. These methods often require manual design of features and may not perform well when dealing with complex scenarios and changing conditions. Therefore, traffic and road signal recognition have witnessed a transformative shift with the advent of deep learning technologies. Convolutional neural networks (CNNs) developed from deep learning has shown prominent capabilities in accurately detecting and interpreting traffic signs and signals from images and videos. In this project, the whole method is mainly based on two parts, which refers to the process of signal-capturing and the detecting process. The CNN model used in the latter part consists of various function layers, including max-pooling layer, linear layer, 2D convolution layer and batch- normalization layer to modify accuracy. This model is trained and tested with Chinese Traffic Sign Dataset, with 58 categories of over 6000 pictures. With the assistance of OpenCV, the model may detect real-time traffic signals and convey messages to vehicles. After adjusting parameters in the CNN model, high probability tags are obtained during the process, with both accuracy and average loss labeled after each epoch. The results demonstrate that the proposed model achieved the testing accuracy of over 95 percent, after undergoing 200 epochs.

Keywords: Deep Learning, Image Recognition, Convolutional Neural Network (CNN).

1. Introduction

Nowadays, with the increasing complexity of urban environments and the growing demand for efficient traffic management, accurate and reliable recognition of traffic signs and signals has become a critical aspect of modern transportation systems [1]. Convolutional neural networks among deep learning methods have developed to be powerful tools for addressing these challenges, offering the capability to automatically learn discriminative features from raw input data and achieve advanced performance in objects detection, classification, and semantic segmentation tasks.

In the previous researches, several researchers have built several models to solve the case. For instance, in Tsoi and Wheelus ‘s project [2], TSR was performed using a cost-sensitive deep learning CNN model with an accuracy of 86.5%. The model contains three convolutional layers, two max-pooling layers, three fully connected layers, two dropout layers, and one softmax layer.Shabarinath and Muralidhar [3] utilized their model to categorize German Traffic Sign Detection Benchmark, and

DOI: 10.54254/2755-2721/76/20240617

eventually reached 99% validation accuracy as well as the test accuracy of 100%, symbolizing the great capacity of CNN models.

Among all traffic signs, various warning signs are the most pivotal section. In the Chinese Traffic Sign Dataset provided on behalf of The Best Road Sign Design Datasets of 2022, most of the signs call for drivers' responses to potential dangers. An intelligent real-time automated traffic sign detection can support drivers and efficient navigation can be provided within self-driving cars. Besides that, the automatic recognition of traffic signs possesses various practical applications in driver assistance systems (DAS). DAS refers to the automatic identification of traffic signs, that is, the detection and classification of traffic signs. In real time situations, traffic sign recognition plays a vital role in unmanned driving. In our work, a CNN-based model is designed for detecting and categorizing 58 discrepant traffic signs. In order to achieve the best performance, several attempts of modifying parameters were tried, and after undergoing different testing accuracy, (varying from 30 percent to 72 percent), eventually the accuracy reached 95.7 percent.

2. Previous Knowledge

2.1. Deep learning

Deep learning is a sub-proposition of artificial intelligence (AI) and machine learning (ML) that aims to mimic the way the human brain works, enabling computers to learn from large amounts of data and make predictions or decisions without being explicitly programmed. It involves the utilization of deep neural networks, which consist of multi-layers of interconnected nodes, to process and extract hierarchical features from complex data. [4]

2.2. Convolutional Neural Networks (CNNs)

CNNs are a type of deep neural network specifically designed to process structured grid data, such as images and video. They use convolutional layers, pooling layers and activation functions to effectively learn hierarchical representations of visual features, thus achieving tasks such as object detection, image classification and semantic segmentation.[5]

Mathematically, the convolution operation can be shown as follows.

$$(I * K)(x, y) = \sum_{i=1}^{m} \sum_{j=1}^{n} I(x - i, y - j) * K(i, j)$$

Where:

- (I*K)(x,y) represents the output value at position(x,y) in the output feature map.
- I(x-i,y-j) represents the pixel value of the input image at position(x - i, y - j).
- K(i,j) represents the weight(or kernel coefficient) of the filter at position(i,j).
- m and n are the dimensions of the filter.

2.3. Convolutional layer

A convolutional layer is a fundamental building block in Convolutional Neural Networks (CNNs), designed specifically to extract features from input data, such as images. [6] It applies a set of learnable filters to the input data to produce a set of feature maps, each capturing different aspects of the input. The key components and operations of a convolutional layer may contain:

2.3.1. In Channels:"in channels" refers to the number of channels in the input data or feature map that are processed by the layer.

2.3.2. Out Channels: Conversely, "out channels" refer to the number of channels in the output feature map produced by the convolutional layer.

DOI: 10.54254/2755-2721/76/20240617

2.3.3. kernel size: the "kernel size" refers to the dimensions of the convolutional filter (also known as the kernel or the feature detector) applied to the input data during the convolution operation.

2.3.4. Stride: The stride parameter determines the step size of the filter as it slides across the input data during convolution.

2.3.5. Padding: "Padding" refers to the technique of adding additional boundary pixels around the input data before applying the convolution operation.

2.3.6. Dilation: "dilation" refers to a technique used to adjust the spacing between the elements (e.g., pixels) of a filter kernel during the convolution operation.

Convolutional layers are typically stacked together in CNN architectures, with multiple layers capturing increasingly abstract and complex features of the input data. [7] This hierarchical feature learning enables CNNs to achieve advanced performance in objects detection, classification, and semantic segmentation tasks.

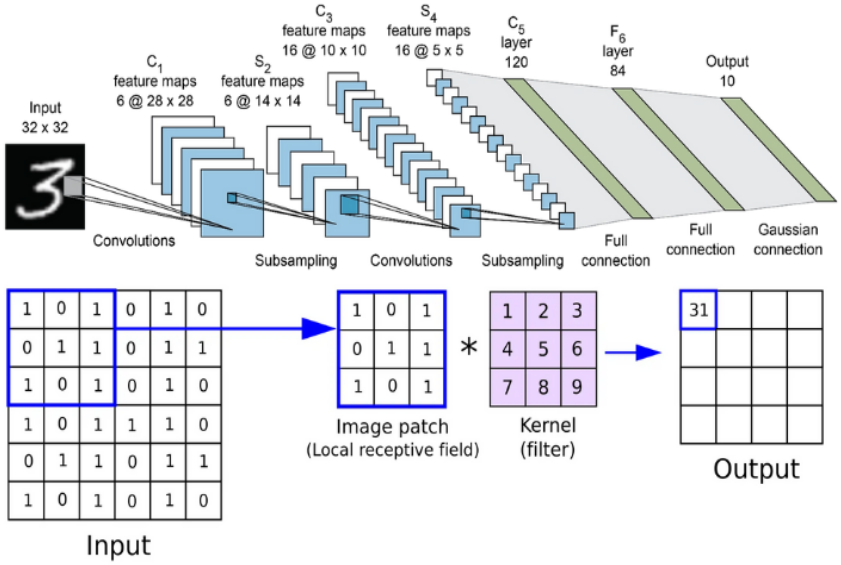

Figure 1. How a convolution layer is utilized

2.4. Activated Functions

2.4.1. ReLU

which stands for Rectified Linear Unit, is a widely used activation function in neural networks, including Convolutional Neural Networks (CNNs). It introduces non-linearity into the model by mapping negative input values to zero, while unchanging those positive input values.

The mathematical expression of ReLU is: $f(x) = \max(0, x)$

2.4.2. Maxpool2d

Maxpooling is a down-sampling operation commonly used in Convolutional Neural Networks (CNNs) to reduce the spatial dimensions of feature maps while retaining important features.

2.4.3. Linear

A linear transformation in a neural network layer involves multiplying the input data by a weight matrix and adding a bias vector.

DOI: 10.54254/2755-2721/76/20240617

2.4.4. BatchNorm1d

BatchNorm1d refers to a specific type of normalization layer used in neural networks, particularly in deep learning frameworks like PyTorch.

2.5. Optimization Algorithm:

Stochastic Gradient Descent (SGD): This is one of the most widely and simplest used optimization algorithms. It updates the model parameters based on the gradient of the loss function with respect to the parameters.

2.6. Loss Function:

nn.EntropyLoss: it is a loss function commonly used in classification tasks when the model is required to output class probabilities. It combines the softmax function and the negative log likelihood loss into a single efficient computation. The mathematical equation is shown as below.

$$CE(x_i, y_i) = -\log(\frac{exp(x_{i,y_i})}{\sum_{j=1}^{C} exp(x_{i,j})})$$

Where:

- x_i is the output of the neural network for example i.
- x_{i,y_i} is the logit for the true class of example i.
- C is the number of classes.
- exp $(x_{i,j})$ computes the exponential of $x_{i,j}$.
- y_i is the true class label for example i.

3. Experiment Method

*Dataset website:*The Best Road Sign Detection Datasets of 2022 | Twine

3.1. Pre-training

In the given dataset, about 6000 pictures are merely unfolded into a big file. So before training, it's necessary to give each category a particular name and categorize them. [8]The relevant code is attached in the given github.This code will roughly categorize these pictures, modify them to the same size and separate them into 2 groups, which are the training group and the testing group, according to the given ratio(in this experiment ratio = 0.2). According to the number of the data set, the first three digits can be used as the group number to extract different group information. Besides that, during this course the test groups can be randomly selected as the test part in all the data sets according to the proportion through the random function.

Figure 2. Samples from the dataset

DOI: 10.54254/2755-2721/76/20240617

3.2. Data-processing

Some of the given pictures are in different sizes, so before data training process, all of these figures should be initialized into the same size. In our project, this is reached by the Reshape and CenterCrop function. [9] Then these figures are transformed into tensors for CNN to detect.

In the project, two functions are defined to separately represent model training and testing. In each function, gradient calculation is disabled during the evaluation phase to speed up the process. These models can predict the categories of traffic signs and give labels to figures. The predicted labels will then be compared to genuine categories. The accuracy of the models is determined by the percentage of the correct quantity. During the course, the rates of train and test loss are also recorded, in order to determine the best model eventually.

An optimizer is added into the training function, so that after each epoch, model is back-propagated due to its loss rate. Parameters in the model are updated when new training batches enroll, and then gradients will be initialized to zero. Accuracy and loss rate are returned.

3.3. Building the CNN model

The whole process including several steps as shown below.

Firstly, the input has 3 channels, which symbolize the RGB colors of the image. In the first convolutional layer, 3 input channels will turn into 12 output channels with a kernel size of 5x5, stride of 2, padding of 0 and dilation of 1. The following equations show the calculation formula of Conv2d.

$$H_{out} = \left\lfloor \frac{H_{in} + 2 \times padding[0] - dilation[0] \times (kernel\_size[0] - 1) - 1}{stride[0]} + 1 \right\rfloor$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2 \times padding[1] - dilation[1] \times (kernel\_size[1] - 1) - 1}{stride[1]} + 1 \right\rfloor$$

Then ReLU function is applied after the first convolutional layer. Secondly, pooling layer is activated with MaxPool2d function. The max pooling layer has a kernel size of 2x2 and stride of 2.

The next operation is the second convolutional layer with 12 input channels (output from the previous layer), 24 output channels, and a kernel size of 5x5. Through these layers, information in the original image will be further extracted and identified to improve the accuracy of recognition.

The third step is another utilization of ReLU and MaxPool2d function to the second convolutional layer. The kernel size remains 2x2, and the stride is still 2.

After that, the flatten function is used to flatten all the dimensions of tensors. These outputs will be modified with Linear function, which contains a fully connected layer with 24 * 47 * 47 input features (output from the prior convolutional layer) and 140 output features. Then Batch normalization layer is applied to the output of the first fully connected layer. ReLU is activated after each of these 2 progresses.

Finally, the fully connected layer with 140 input features (output from the batch normalization layer) will undergo the last linear course and turn into 58 output features, which represent the 58 categories of traffic signs.

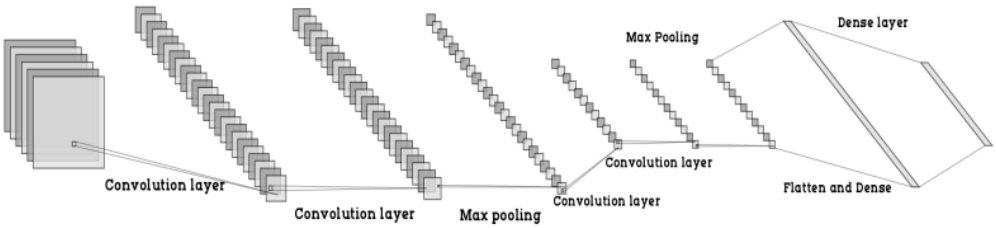

Figure 3. Discription of the CNN model

DOI: 10.54254/2755-2721/76/20240617

3.4. Parameters in the model:

After several attempts, these parameters best match with the eventual consequence. Using these parameters may achieve an approximate detecting rate of 95%.

Table 1. Parameters used in the CNN model

Parameters	Batch size	Learning rate	Epochs	Optimizer	Size of figures
Value	64	1e-3	200	Stochastic Gradient Descent	200 x 200

The categories of traffic signs can be seen in the code.

4. Results

The loss rates and accuracy of both training and testing progresses are recorded into lists, and then plotted into a whole picture. It shows the initial consequences of the first 100 epochs.

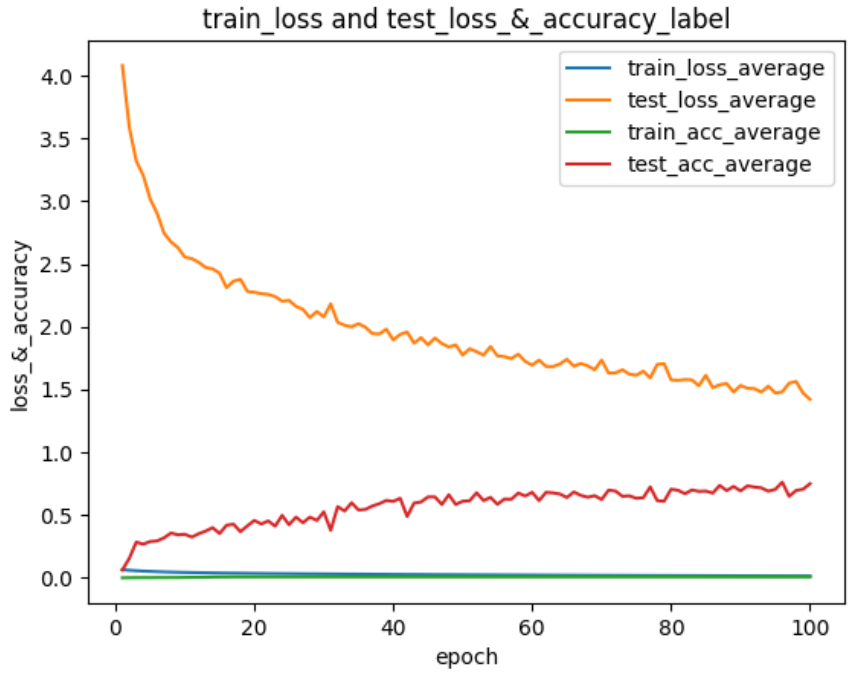

Figure 4. Consequence of 100 epochs

According to the figure, both of the accuracy and loss rate in the training progress present in a low level. The conclusion shows the feasibility of the built training model. However, the loss rate in testing progress is about 1.5, higher than expected. This may be the consequence of under-fitting, and the accuracy of testing is merely around 75%.

When adjusting the epoch number to 200, the test_loss_average decreased to 0.46, and the test accuracy reached 95.7 percent, which matched with the expectation.

DOI: 10.54254/2755-2721/76/20240617

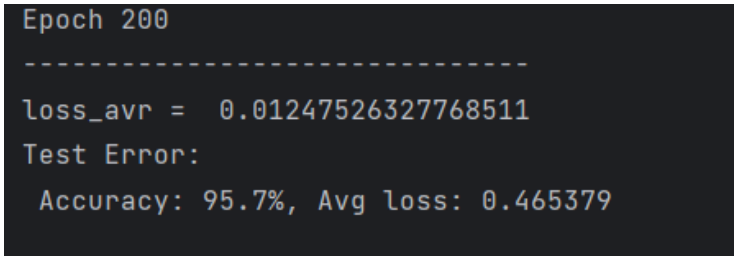

Figure 6. Consequences of 200 epochs

All of the codes can be viewed on the website given below.
https://github.com/jiangwei5288/A-Traffic-and-Road-Signal-Recognition-Method-through-Deep-Learning

5. Conclusion

In the experiment, the accuracy surpassed the expected 95%. With several attempts to optimize the model, the accuracy and loss rates have presented marked declines. However, in real driving and detecting process, there's much for further improvement. Among the influencing factors, the decisions of each parameter may vary a lot, and there might be other factors that aren't concerned in this experiment. While in the training case, the average loss of each graph tends to be little, the average loss in testing remains higher. By adjusting parameters in the model so far, no better results have emerged. On the other hand, due to the lack of identification and detection of actual traffic signs in the experiment, the results of this model on the actual situation are not clear yet. In future related works, this model will be tested with OpenCV cameras as well.

6. Future Work

6.1. Real-time Signal Recognition for Adaptive Cruise Control (ACC)

Integrate CNN model with adaptive cruise control systems to enable vehicles to automatically adjust their speed and distance from other vehicles based on recognized traffic and road signals. For example, the vehicle can automatically accelerate, decelerate, or maintain its speed in response to speed limit signs, traffic lights, and stop signs.

6.2. Driver Assistance and Alerts:

Integrate CNN model with driver assistance systems to provide real-time alerts and assistance to drivers. For example, the system can alert the driver when they are approaching a stop sign, red light, or pedestrian crossing. It can also provide warnings if the vehicle is drifting out of its lane or if there is an obstacle in the road.

6.3. Emergency Vehicle Detection and Response:

Train CNN model to detect emergency vehicles, such as ambulances and fire trucks, and prioritize their passage through traffic. Vehicles equipped with your CNN model can automatically yield to emergency vehicles, clear a path for them, and adjust their speed and trajectory to facilitate rapid response and emergency services.

References

[1] Zhang, Y., Song, S., Tan, W., & Liu, Y. (2021). Real-Time Traffic Sign Recognition Based on Improved YOLOv3. IEEE Access, 9, 104815-104825. [DOI: 10.1109/ACCESS.2021.3084623]

[2] T. S. Tsoi and C. Wheelus, "Traffic Signal Classification with Cost-Sensitive Deep Learning Models," in 2020 IEEE International Conference on Knowledge Graph (ICKG), Aug. 2020, pp. 586–592

DOI: 10.54254/2755-2721/76/20240617

[3] B. B. Shabarinath and P. Muralidhar, “Convolutional Neural Network based Traffic-Sign Classifier Optimized for Edge Inference,” in 2020 IEEE REGION 10 CONFERENCE (TENCON), Nov. 2020, pp. 420–425, iSSN: 2159-3450.

[4] Pramod Sai Kondamari, Anudeep Itha, ”A Deep Learning Application for Traffic Sign Classification”

[5] S. Qian, H. Liu, C. Liu, S. Wu, and H. San Wong, “Adaptive activation functions in convolutional neural networks,” Neurocomputing, vol. 272, pp. 204–212, 2018.

[6] Zhang, Y., Yang, J., Lu, J., & Zhao, J. (2016). Traffic sign detection and classification in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[7] Jain, H., & Agrawal, N. (2016). A review of traffic sign detection and recognition systems. arXiv preprint arXiv:1606.09518.

[8] Huang, C., Li, Y., Loy, C. C., & Tang, X. (2017). Learning deep representation for imbalanced classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[9] Zhang, Z., Zhang, C., Shen, W., & Zhang, Y. (2021). Road Sign Recognition Based on Improved YOLOv3-Tiny Model. IEEE Access, 9, 17964-17975. [DOI: 10.1109/ACCESS.2021.3060145]