PROJECT REPORT

CMPE272 -ENTERPRISE SOFTWARE PLATFORMS

GROUP 12

Instructor: Rakesh Ranjan

Spring 2020

TRAFFIC SIGN RECOGNITION USING CONVOLUTIONAL NEURAL NETWORKS

SAN JOSÉ STATE UNIVERSITY

Sreeja Sriram
Software Engineering
San Jose State University
San Jose, CA
sreeja.sriram@sjsu.edu

Harshit Malwiya
Software Engineering
San Jose State University
San Jose, CA
harshit.malwiya@sjsu.edu

Poojitha Vaddey
Software Engineering
San Jose State University
San Jose, CA
kanakalakshmipoojitha.vaddey@sjsu.edu

Shweta Mane
Software Engineering
San Jose State University
San Jose, CA
shwetaashokrao.mane@sjsu.edu

Traffic Sign Recognition Using Convolutional Neural Networks

Sreeja Sriram
Software Engineering
San Jose State University
San Jose, CA
sreeja.sriram@sjsu.edu

Harshit Malwiya
Software Engineering
San Jose State University
San Jose, CA
harshit.malwiya@sjsu.edu

Poojitha Vaddey
Software Engineering
San Jose State University
San Jose, CA
kanakalakshmipoojitha.vaddey@sjsu.edu

Shweta Mane
Software Engineering
San Jose State University
San Jose, CA
shwetaashokrao.mane@sjsu.edu

I. INTRODUCTION

In the world of Artificial Intelligence and advancement in technologies, many researchers and big companies like Tesla, Uber, Google, Mercedes-Benz, Audi, etc are working on autonomous vehicles or self-driving cars. For achieving accuracy in this area, it is necessary for vehicles to understand and follow all traffic rules. The vehicles should be able to interpret traffic signs and make decisions accordingly. Under this project, we have built a machine learning model that takes an image of a traffic sign and predicts it's meaning.

Traffic signs are an integral part of our road infrastructure. They provide critical information for road users, which in turn requires them to adjust their driving behavior to make sure they adhere to whatever road regulation is currently enforced. Autonomous vehicles must also abide by road legislation and therefore recognize and understand traffic signs. There are several different types of traffic signs like speed limits, no entry, turn left or right, children crossing, no passing of heavy vehicles, etc. Traffic signs classification is the process of identifying which class a traffic sign belongs to. Using the most popular image classification strategy Convolutional Neural Network, we have built a machine learning model that classifies traffic signs present in the image into different categories.

The traffic sign classification and recognition experiments are conducted based on the German Traffic Sign Recognition Benchmark. The favorable prediction and accurate recognition of traffic signs are achieved through the continuous training and testing of the network model. Experimental results show that the accurate recognition rate of traffic signs reaches 95.13%. Compared with other algorithms, the proposed algorithm has remarkable accuracy and real-time performance, strong generalization ability, and high training efficiency.

II. RELATED WORK

The last decade shows a growth evolution in the development of intelligent transportation systems (ITS) and especially ADAS and Self-Driving Cars (SDC). In these systems, traffic signs detection and recognition is one of the difficult tasks that confront researchers and developers. This issue is addressed as a problem of detecting, recognizing, and classifying objects (traffic signs) using computer vision and still be a challenge until now. The work presented in this paper focuses on traffic signs recognition without the consideration of the detection step. For this purpose, this section discusses only related works from this angle. Traffic signs recognition is divided in two parts: features extraction and sings recognition.

The authors proposed ways in which the traffic sign can be effectively detected based on the shape features and how the HSV color space is used for partial threshold segmentation and the recognition of the images using convolutional neural networks by adding the batch normalization processing after the pooling layer and selecting Adam method as the optimizer algorithm in [1]. The presented method in [2] uses a modified LeNet-5 network to extract a deep representation of traffic signs to perform the recognition. It is constituted of a Convolutional Neural Network (CNN) modified by connecting the output of all convolutional layers to the Multilayer Perceptron (MLP). In [3], several methods have been proposed, including edge detection.

III. TRAINING AND TEST DATA

The training dataset contains images of traffic signs from German traffic sign recognition benchmarks (GTSRB) dataset belonging to 43 unique classes. The traffic signs are symbolic representations for no passing, yield, no entry, double curve, etc.

Fig 1. Training images from GTSRB Dataset

In order to validate the model, the training data has been split into 80% training and 20% validation

IV. MACHINE LEARNING MODEL

Among various network architectures used in deep learning, convolutional neural networks(CNN) are widely used in image recognition. CNNs consist of convolutional layers, which are sets of image filters convoluted to images or feature maps, along with other (e.g., pooling) layers. In image classification, feature maps are extracted through convolution and other processing layers repetitively and the network eventually outputs a label indicating an estimated class.

Convolutional Networks are a category of Neural Networks that have proven very effective in areas such as image recognition and classification. ConNets have been successful in identifying faces, objects, and traffic signs apart from powering vision in robots and self-driving cars.

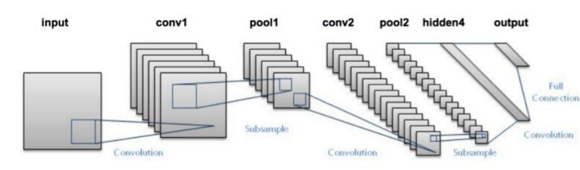

Fig 2. CNN Model Architecture

ConvNets derive their name from the ”convolution” operator. The primary purpose of Convolution in the case of a ConvNet is to extract features from the input image. Convolution preserves the spatial relationship between pixels by learning image features using small filters.

The Convolutional Neural Networks was chosen for the following reasons:

1. CNN does not require human supervision for feature selection.

2. When we compare handcrafted features with the CNN model, CNN’s performance is good and it gives better accuracy.

3. If we use a fully connected layer to extract the features, the input image of size 32x32 and a hidden layer having 1000 features will require an order of 10pow6 coefficients, a huge memory requirement. In the convolutional layer, the same coefficients are used across different locations in the space, so the memory requirement is drastically reduced.

4. By using the standard neural network which is equivalent to a CNN, as the number of parameters would be much higher, the training time would also increase proportionately. In CNN, since the number of parameters is drastically reduced, training time is proportionately reduced.

There are many important parts of the Convolution Network. These include the following properties

1. Depth: Depth corresponds to the number of filters we use for the convolution operation.

2. Stride: Stride is the number of pixels by which we slide our filter matrix over the input matrix.

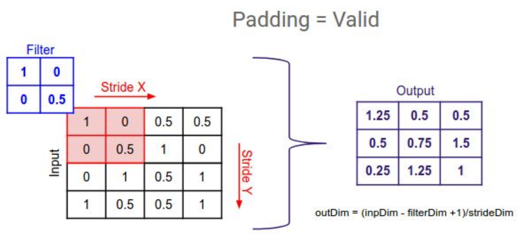

Fig 3. Convolution filter

3. Zero-padding: Sometimes, it is convenient to pad the input matrix with zeros around the border, so that we can apply the filter to bordering elements of our input image matrix.

4. Non-Linearity: ReLU is an element-wise operation (applied per pixel) and replaces all negative pixel values in the feature map by zero. The purpose of ReLU is to introduce non-linearity in our ConvNet, since most of the real-world data we would want our ConvNet to learn would be non-linear (Convolution is a linear operation element-wise matrix multiplication and addition, so we account for non-linearity by introducing a nonlinear function like ReLU).

5. Spatial Pooling: Spatial Pooling (also called subsampling or downsampling) reduces the dimensionality of each feature map but retains the most important information. Spatial Pooling can be of different types: Max, Average, Sum etc.

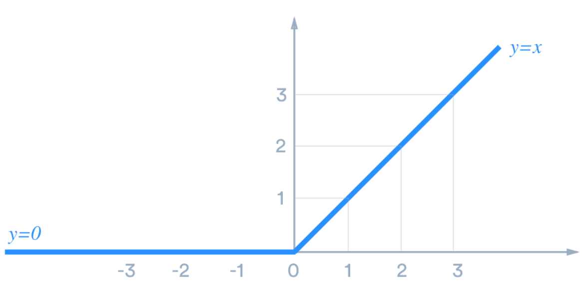

Fig 4. ReLU activation function

These networks have grown in the number of layers leading to architectures such as ResNet and AlexNet that have been trained on images such as Cifar-10 and then fine-tuned to other problems.

V. ARCHITECTURE

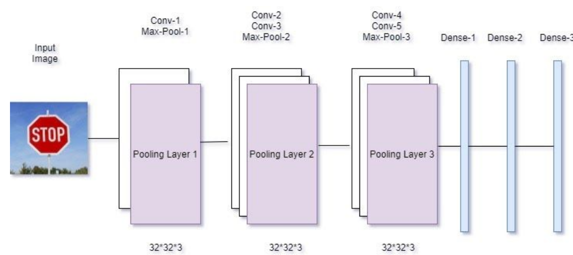

Fig 5. Model Architecture

EXPERIMENT

The algorithm involves extracting the features from the input images. The images are processed through multiple convoluted layers to obtain the final result.

1. In the Convolution layer, the input images of size 32*32 are processed as tensors. Tensors are matrices of numbers with additional dimensions on which convolution is performed. In our model, we have 5 convolution layers A n*n matrix called a 'filter' or 'kernel' or 'feature detector' is used for performing convolution. In our model, for the first layer, the convolution is performed by a kernel of size 5*5 and for the remaining layers, a 3*3 filter is used. A matrix is formed by sliding the filter over the image and computing the dot product is called the 'Convolved Feature' or 'Activation Map' or the 'Feature Map'. A filter slides over the input image (convolution operation) to produce a feature map. The convolution of another filter, over the same image gives a different feature map.

2. Spatial Pooling reduces the dimensionality of each feature map but retains the most important information. Spatial Pooling can be of different types: Max, Average, Sum etc. In our model, max pooling is performed on the feature maps with grid size of 2*2.

3. An additional operation called ReLU has been used after every Convolution operation. ReLU is an element-wise operation (applied per pixel) and replaces all negative pixel values in the feature map

by zero. The purpose of ReLU is to introduce non-linearity in the ConvNet.

4. The Fully Connected Layer uses a softmax activation function in the output layer. The no. of dense layers in this layer is 3. In this layer, we are using a softmax function for introducing non-linearity. The output from the convolutional and pooling layers represent high-level features of the input image. The purpose of the Fully Connected layer is to use these features for classifying the input image into various classes based on the training dataset.

VI. RESULT

The following are the results obtained from the classification of traffic signs using the above model.

Model	**Accuracy (%)**
CNN	95.13

Table 1. Model accuracy table

The following are the training and validation accuracy and loss graphs for the model.

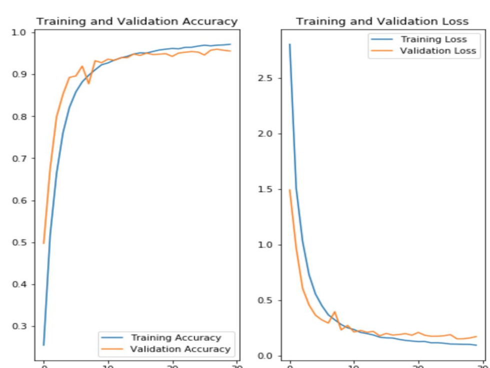

Fig 6. Model accuracy and loss trends

VII. REFERENCES

[1] D. Yasmina, R. Karima and A. Ouahiba, "*Traffic signs recognition with deep learning,*" 2018 International Conference on Applied Smart Systems (ICASS), Medea, Algeria, 2018, pp. 1-5, doi: 10.1109/ICASS.2018.8652024.

[2] Yasmina, Djebbara & Karima, Rebai & Azouaoui, O.. (2018). *Traffic signs recognition with deep learning*. 1-5. 10.1109/ICASS.2018.8652024

[3] P. Dewan, R. Vig, N. Shukla and B. K. Das, "An Overview of Traffic Signs Recognition Methods," International Journal of Computer Applications, Vol. 168 – N..11, June 2017

[4] D. Jianmin and V. Malichenko, "Real time road edges detection and road signs recognition," IEEE International Conference on Control, Automation and Information Sciences (ICCAIS), Changshu, China, 29- 31 Oct. 2015