Traffic Sign Classification

Koushik Reddy Parukola,
Department of Computer Science and Engineering,
New Horizon College of Engineering,
Bangalore, India

Abstract:- Traffic Sign Recognition is an intelligent system built based on computer vision and neural networks for detecting, classifying and recognizing traffic signs. This system is so accurate in detecting the signs given as it is trained using the power of deep learning. This system requires huge amount of data for good and accurate output. In this system there are 43 classes of traffic signs with thousands of images, and this system is pre trained using convolutional neural networks (CNN) which has accuracy above 90% in detecting images. After training the system images will be given using the camera as input, which are further preprocessed in different ways which are then passed to the convolutional neural network for detection. The final result is given to the user with the accuracy above 95% along with the name of the sign with the help of data classes the data classes.

Keywords:- *Traffic Sign Detection, Deep Learning, Computer Vision, Convolutional Neural Network, Numpy, Keras, Tensorflow.*

I. INTRODUCTION

The identification of traffic signs is an important research area in computer vision and an important component of the Advanced Driver Assistance System. There is a very vast increase in the count autonomous cars which has no need of human intervention for driving. This system can be differentiated in two ways namely traffic sign recognition and traffic sign detection. The final result from the system will be the traffic sign recognition which is directly influenced by the accuracy at which the traffic sign is detected. Traffic signs convey import messages about vehicle safety, such as Traffic conditions, road rights, forbidding, driver directions, cautions on dangers and the other things.

There are some certain features that can help drivers to acquire road information which are color and shape, so they are some of the characters that can be used for traffic sign identification and classification. Almost all the countries traffic signs are similar in terms of color (red, blue, yellow) and shapes (circle, square, triangle). [1]This paper proposes a method that uses photographs to detect and distinguish various forms of traffic signs. This paper varies from others in that it uses internationally recognized signs rather than a small number of signs like many other papers do. For detection and recognition, two different neural networks were used; one classifies the symbol, while the other classifies the form.

The sign classifier tries to classify the sign, while the form classifier tries to classify the object's shape. Finally, the effects of both classifiers are used to determine if the picture patch includes a traffic sign or not. Convolutional neural network was chosen for this goal because it outperforms SVM. In paper [2] Hough Transform is used to detect and pre-process the road traffic signs before recognized, which greatly helps to improve the accuracy and timeliness. To obtain more information and improve detection efficiency, a multi-resolution feature combination network fabric is created, which is capable of studying many useful features from small artefacts. Additionally, the traffic sign detection system is divided into spatial sequence classification and regression tasks. This paper [3] proposes a full traffic sign recognition pipeline that outperforms the majority of previous work in terms of generality, reliability, and run time. The proposed detector and classifier can be trained from start to finish and are capable of real-time applications. The proposed detector has been shown to be capable of detecting traffic signals that are not limited by color or shape. And the proposed classifier has been shown to be both simple and efficient. And this paper said that the improvement in the traffic signs is most important. And they also gave a view that the usage of more annotated data will boost the performance of the system.

Fig 1: A Hair Pin Bend Traffic Sign [5]

II. RELATED WORK

A. *Pre-Processing*

The task of traffic sign detection is very challenging due to quite a few disturbances, such as nonuniform illumination, motion blur, occlusion and hard negative samples. However, as preprocessing of the image before the image is sent to the neural networks is done regarding all those issues like blurriness, converting to grayscale, cleaning the image for noise reductions. This pre-processing is the not the same in all approaches. It should be done according the input given from user and the input for further processing.

So, different papers proposed their different ways of pre-processing input images. For example, in paper [4] for pre-processing the first step in classifying any item is to identify the appropriate area. There are several methods for determining the best return on investment. To find regions that may contain a traffic sign, they used maximally stable extremal regions (MSERs). Each frame is binarized at a different number of threshold levels, and connected components are found at each level. MSERs are components that retain their shape over a number of different threshold levels. Different color enhancement techniques have been used to keep the traffic signs' shapes within the threshold stages.

B. *Traffic Sign Classsification*

As the recognition of traffic sign is the core part in the system there are a greater number of ways of implementation. Even the implementation is different in technique used is based on deep learning and the approach is by creating neural networks. So, all the related work in maximum number of papers is done using these neural networks. So next the image classification has to be done according to the output given from the neural networks.

III. METHODOLOGY

C. *Pre-Processing*

Not only in the case of this traffic sign recognition system but all the other computer vision related systems need their input image to be pre-processed into the convenient way for further classification. In this case of traffic sign classification pre-processing the input data is very important because the image taken itself may not be with exact same dimensions or features in terms of high, width and other categories. So, this made the pre-processing of the input image must in the process. The image given for the system as input may not be same in dimension’s, size of the image, so in this case we expand the image or diminish the size of the image as out requirement.

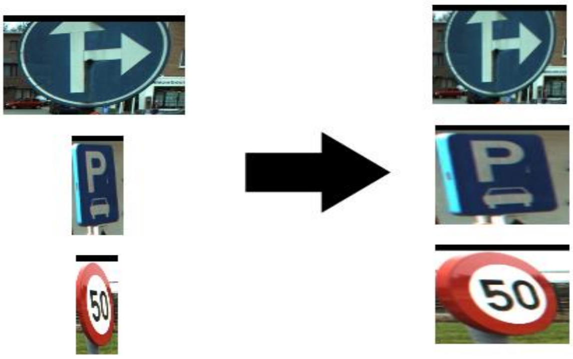
Fig 2: Image Resizing [6]

After the image altering is done our image will be ready for giving as input for the neural network but the image itself cannot be given to the model even it is in required dimension’s because neural networks work on matrices with values not on images itself. So, the pre-processed images are first converted to a NumPy array then the NumPy array is given to the neural network.

D. *Convolutional Neural Network*

A convolutional neural network namely ConvNet or CNN is a deep learning algorithm which takes input in the form of an image and assign importance like weights for different features of the image which will further be used to classify or differentiate images from one another. Using the convolutional neural network also reduces the need of pre-processing the image but not eliminating the need to pre-process the image. Images are nothing but a matrix which usually contains the information of pixels. This neural network is built by several layers, each with its unique task for image recognition or classification. And the architecture of these neural networks resembles the human brain. In brain there will be neurons connected to each other passing information from one another, the same happens in the case of these neural networks. The main role of the convolutional neural network is to reduce the images info a form which is much simple for processing without losing any features of the image.

The first layer of the neural network is the convolutional layer. This layer has the most of user specified parameters such as size of kernel and number of kernels. The main objective of this layer is feature detection from the image, those features are here are feature maps.

With the convolutional layer there is an activation used named as RELU (Rectified Linear Unit). This activation uses the rectifier function for increasing the non-linearity in the images while capturing the features maps. The main reason for choosing this activation function upon sigmoid is that this converts all the negative values to zero thus helping in better classification.

RELU: $f(x) = \max(0, x)$

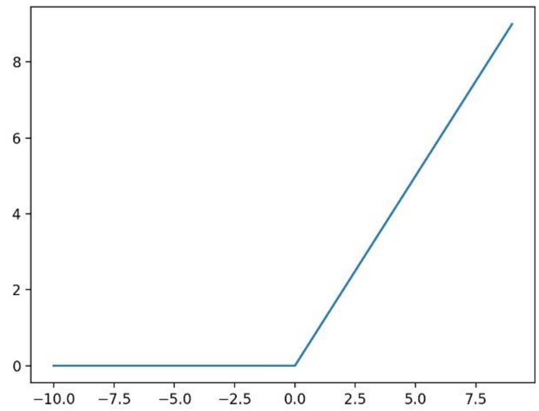

Fig 3: Graph of RELU [7]

Pooling is the next layer following convolutional layer in the neural network. The work of this pooling layer is to reduce the dimensions of the feature maps. This will be done without losing any of the features extracted from the image. There are different types in pooling and the most commonly used one is the max pooling or maximum pooling which preserves the maximum value means it will preserve the most highlighting feature.

The next is a small layer which is known as the flattening layer which converts the pooled map into a 1-dimentional array and then is connected into a fully connected layer.

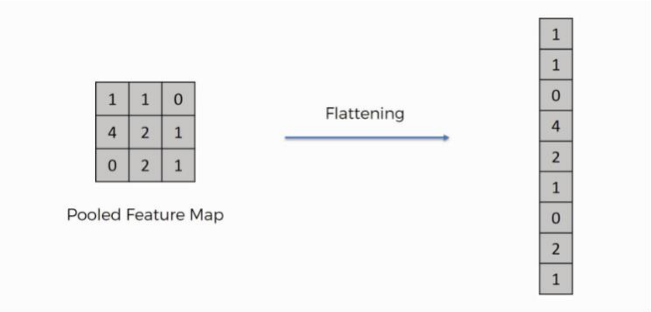

Fig 4: Flattening Layer [8]

The output from the flattening layer is sent to the fully connected layers which forms more attributed with filters and helps in better classification. A function named SoftMax is used at the end of the process for normalizing the output of the neural network to a probability distribution over the output classes.

IV. SYSTEM OVERVIEW

The system which is built upon the convolutional neural networks follow the same procedure of image pre-processing, image detection and classification. But the way in which pre-processing or image detection is done varies depending on the type of images also.

E. *Training*

For training the model first we need to find a suitable data set which has huge amount of data in our case images. Next importing the imaged has to be done along with resizing the image and conversion of that image to a NumPy array.

Once the image is given as the input to the system, the image dimensions and sizes are transformed according to the further usage. The image resizing is done using resize method in CV2 library and the resizing image is done using NumPy expand dimensions method. Once it is done as the next Step the image is converted into a NumPy array as the neural networks will process the image according to the image matrix which is obtained by the array from the image. We know that the model trained should also be tested, so we need some part of the data for testing which we obtain by splitting the training data with the size of your choice.

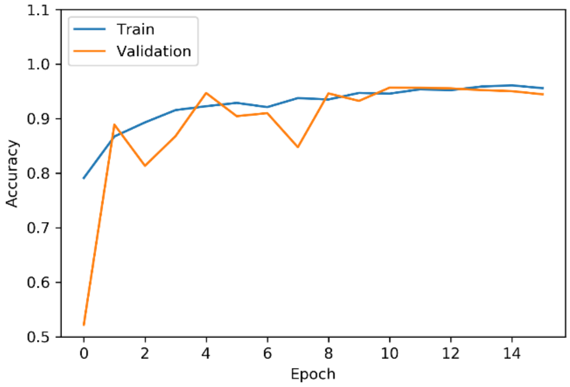

Fig 5: Accuracy Rate Curve [9]

After that we set the number of epochs and train the model with the data given. The term Epoch means the number of times the batch should run or in general terms we can say how many times should the training process should happen. More the number of epochs more the accuracy and also more the epochs more the time taken for training and testing.

F. *Classification*

After all the training and the testing process the it is ready for usage. We can input an image with the needed pre-processing done and expect the classification done by the model as we pass the NumPy array to the model for predicting the class.

An extra feature added for this system is the voice statement given by the system to the user according to the predicted class by the model. The text to voice is done by the library GTTS (google text to speech) and the sound is played using playsound method from playsound library. All the voice instructions are saved as strings according to the classes of the signs.

V. FUTURE WORK

This traffic sign classification system is implemented using the CNN (Convolutional Neural Networks) is very fast at processing the images, this can be implemented into a real-life system which will recognize live traffic signs, classify them and instruct the driver through voice. And also, the accuracy can be increased by further improvements in the neural network and image pre-processing.

VI. CONCLUSION

The system proposed will detect and classify a set of 43 different traffic signs. The accuracy of the classification is moderate and the loos rate of the model is low and this accuracy can also be improved by further improvements in the convolutional neural network with also the real time detection or live detection of the implemented as the future work.

REFERENCES

[1]. Md Tarequl Islam, “Traffic sign detection and recognition based on convolutional neural networks,” IEEE 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), 16 March 2020.

[2]. Ying Sun, Pingshu Ge, Dequan Liu, “Traffic Sign Detection and Recognition Based on Convolutional Neural Network,” 2019 Chinese Automation Congress (CAC), 13 February 2020.

[3]. Jia Li, Zengfu Wang, “Real-Time Traffic Sign Recognition Based on Efficient CNNs in the Wild,” IEEE Transactions on Intelligent Transportation Systems (Volume: 20, Issue: 3, March 2019).

[4]. Jack Greenhalgh, Majid Mirmehdi, “Real-Time Detection and Recognition of Road Traffic Signstle of paper if known,” unpublished.

[5]. Image: Hair Pin Bend Traffic Sign, Author: Malay Sautya, Article: “100+ All Traffic Signs or Road Safety Signs in India as per IRC”, source:“https://civilnoteppt.com/traffic-signs-or-road-signs-in-india”.

[6]. Image: Image Resizing, Author:Waleed Abdulla, Source: https://medium.com/@waleedka/traffic-sign-recognition-with-tensorflow-629dffc391a6.

[7]. Image: RELU Graph, Author: Jason Brownlee, Article: A Gentle Introduction to the Rectified Linear Unit (ReLU), Source: “https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks”.

[8]. Image Flattening layer, Author: SuperDataScience Team ,Article: “Convolutional Neural Networks (CNN): Step 3 - Flattening”, Source: “https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-3-flattening”.

[9]. Image: Accuracy Rate Curve by Nathaniel JermainR, Article: “Building a Custom Convolutional Neural Network in Keras”, source:“https://opendatascience.com/building-a-custom-convolutional-neural-network-in-keras”.