International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 09 | Sep 2021 www.irjet.net p-ISSN: 2395-0072

Traffic Sign Recognition and Classification using CNN

Thunga Saiteja[1], Katam Rishik Kumar[2]

[1]Undergrad Student, School of Computer Science and Engineering, VIT University, Vellore, Tamil Nadu, India
[2]Undergrad Student, School of Computer Science and Engineering, VIT University, Vellore, Tamil Nadu, India

---***---

Abstract - *Traffic sign recognition and classification are useful in many ways such as in automated driving cars. While promising results are achieved within the areas of traffic-sign recognition and classification, few works have provided simultaneous solutions to those tasks for world images. The dataset used is the German Traffic Sign Recognition Dataset (GTSRB) which contains almost 40,000 images of different traffic signs which are further classified into 43 different classes. The dataset is quite varied, with some classes having many images while other classes have few images. The dataset has two folders named train which contains the images classified into 43 classes and are used for training of our model. The second folder is the test folder which contains images of traffic signs in different conditions which are used for testing the model. Our core idea is to use CNN to classify traffic signs to perform efficient and accurate traffic sign detection and recognition.*

***Key Words*: German Traffic Sign recognition dataset (GT-SRB), Convolutional Neural Network (CNN), Recognition, classification, Dataset**

1.INTRODUCTION

The recent technical development in mobile processors led many automobile manufacturers to deploy computer vision systems into their cars. These systems ensure safety which is a crucial aspect for autonomous vehicles. Traffic sign recognition is one of the most well-known and widely discussed tasks that can be solved by these computer vision systems. It can ensure safety as it allows the vehicle to know what different signs on the road mean and act accordingly. However, such systems come with problems like low accuracy in detection, and some systems are unable to detect traffic signs from different countries.

2. LITERATURE REVIEW

Traffic sign recognition is generally carried out in two steps: localization and following classification. Many localization methods were proposed [1], [2], [3]. In other papers effective real time implementations of image pre-processing and traffic signs localization algorithms are proposed [4],[5]. Using Generalized Hough Transform (GHT) algorithm allows to find the exact coordinates of a traffic sign in the input image. Thus, the simple template matching technique is used in the classification stage. The training and testing of the developed algorithms were done by using datasets from GTSRB [6]. The testing of the developed technology for detecting and classifying traffic signs for real life conditions, showed a significant decrease in efficiency. Studies have shown that certain factors are responsible the decrease in efficiency like variations in the illumination, contrast, and rotation angle in acquired images of traffic signs. Hence, high quality recognition cannot be achieved by using simple algorithms like template matching due to a limited set of predefined templates. System performance can be improved by combining localization algorithm having good results with recognition using the convolutional neural networks which have received a wide range of applications in recent years

Effective implementation of algorithm for noise removal using CUDA is shown in paper [10]. In GPUs, the acceleration is 60-80 times when compared with conventional execution on a CPU and 1920x1080 pixels is the frame size. The requirements for video processing in real time can be met by using the CUDA-enabled mobile GPU NIVIDIA Jetson TKI in which pre-processing of one videoframe can be done within 7-10ms.

Paper [5] addresses the algorithms for traffic signs tracking and detection. The localization method developed is a modification of the GHT while considering the constraints on time for processing of a single frame. Using the value of present speed of vehicle for tracking improved the performance of the system, as the search area between the adjacent frames can be reduced. The final step, classification ensures that whole procedure is carried out properly.

3. METHODOLOGY PROPOSED

We build a CNN model that classify the traffic sign images into their respective categories. This model enables us to detect and understand traffic signs which plays a crucial part in autonomous vehicles.

Fig 1. Shows the work flow of the system, we first process the data and build a CNN model and then we train and implement it.

We are using a self-made dataset using images from GT-SRB with train and test folders for training and testing the model. The train folder consists of 43 different classes each representing a different traffic sign. The OS module is used to iterate over all the classes and add the images to a list with their respective labels. We use the PIL library to open the image content in the form of array. These lists are converted into NumPy arrays to feed them to the model. The images are reshaped to make them compatible with the model

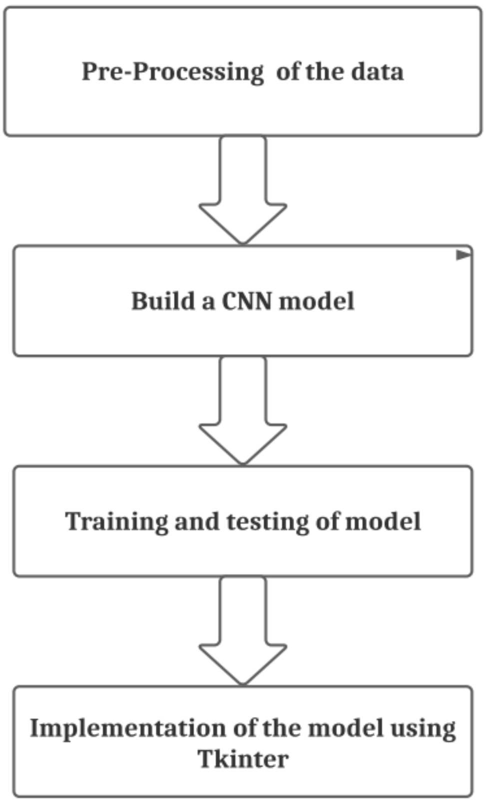

Fig-1: Block diagram

Table-1: ARCHITECTURE OF THE MODEL

2 Conv2D layer
MaxPool2D layer
Dropout layer
2 Conv2D layer
Maxpool2D layer
Dropout layer
Flatten layer to squeeze the layers into 1 dimension
Dense fully connected layer
Dropout layer
Dense layer

The above architecture of our model consists of 4 convolutions layer each 2 of them followed by Maxpooling layers with a stride of 2. It has one fully connected layer, 3 dropout layers and a dense layer. It has a flatten layer to squeeze the layers into one dimension.

The model is compiled by Adam optimizer and is then trained by using the data from dataset. It has shown better results for batch size 64 rather than for batch size 32. The accuracy was stable after 15 epochs with a value of 95%.

From the Fig 3. and Fig 4. we can say that accuracy has been gradually increasing with each epoch whereas the loss has been gradually decreasing with each epoch.

Implementation of the model: We have used Tkinter to create a Graphic User Interface (GUI) to implement our traffic sign classifier. The interface consists of a upload image button which when clicked opens a window to browse the files for selecting a image. The classify button is enabled when the selected image is uploaded using upload button. Upon clicking the classify button, the respective name of the selected traffic sign image is displayed on top of the image. In Fig 5. we can see a GUI with a headline of Traffic sign classification

4. RESULTS:

We got 90% accuracy for 10 epochs while training, then we changed the number of to epochs to 15 and have achieved an accuracy of 95.09% while training the dataset and achieved an accuracy of 95% while testing. The model we made is capable of classifying the images with low light and even some slightly unclear images accurately.

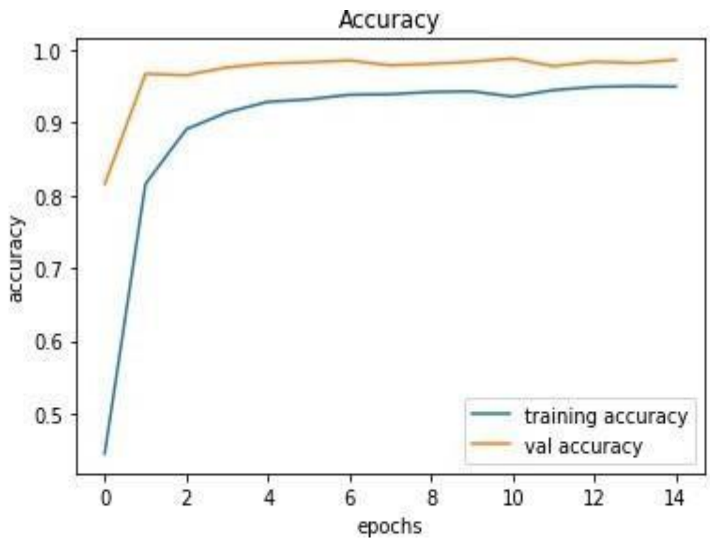

Fig-2: Classification accuracy changing with training iterations

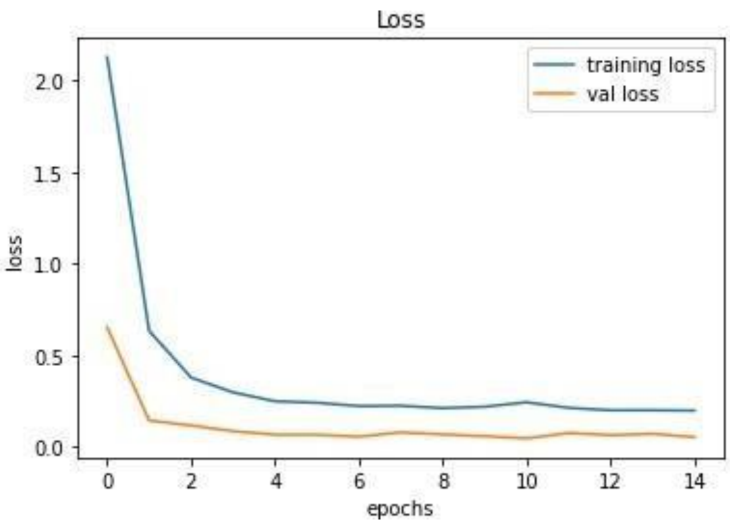

Fig-3: Loss changing with training iterations

Table-2 ACCURACY AND LOSS OF THE MODEL FOR 15 EPOCHS

Number of epochs	Accuracy (%)	Loss
1	38.44	2.4223
2	71.99	0.9316
3	81.26	0.6200
4	85.40	0.4851
5	87.03	0.4313
6	88.98	0.3748
7	90.19	0.3336
8	91.02	0.3062
9	91.53	0.2885
10	92.73	0.2485
11	92.33	0.2680
12	92.65	0.2567
13	93.31	0.2425
14	94.55	0.2314
15	95.09	0.2325

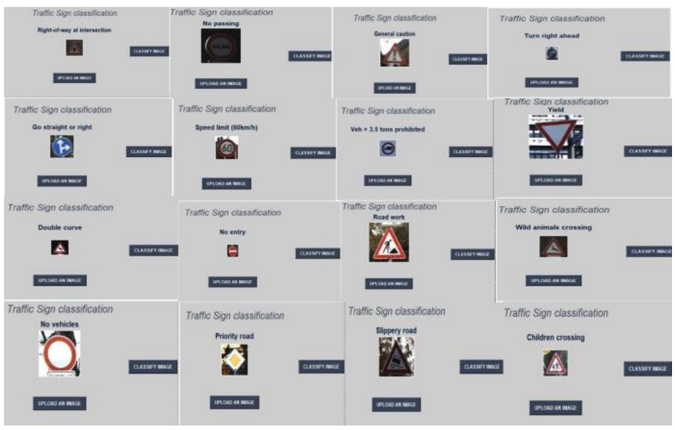

Fig-4: Classification results for different traffic signs

5. CONCLUSION

This project considers associate degree implementation of the classification algorithmic program for the traffic signs recognition task. The proposed model for traffic signs classification shows superb results: 95% accuracy. The planned technique was be a deep learning approach that use CNN model with keras layers. Since the challenge to classify image isn't so high our model even predicts the photographs that aren't a region of the GTSRB quite intuitively. As an example - old Crossing appearance loads like children's crossing and therefore the model will do an honest job predicting it. Color unchangeability is kind of well established, the prediction of flip left ahead works well. For pictures that are not 32x32x3 the size method's order determines however correct the prediction is. The size technique has the tendency to destroy the facet ratios within the image that causes deterioration within the performance of the model. Despite these tiny setbacks our model has the accuracy of about 95%.

6. REFERENCES

[1] Nikonorov, Artem, P. Yakimov, and P. Maksimov. "Traffic sign detection on GPU using color shape regular expressions." *VISIGRAPP IMTA-4* 2013 (2013).

[2] Belaroussi, Rachid, et al. "Road sign detection in images: A case study." *2010 20th International Conference on Pattern Recognition*. IEEE, 2010.

[3] Ruta, Andrzej, et al. "A new approach for in-vehicle camera traffic sign detection and recognition." (2009).

[4] Fursov, Vladimir Alekseevich, Sergey Alexeevich Bibikov, and Pavel Yurievich Yakimov. "Localization of objects contours with different scales in images using Hough transform." *Computer optics* 37.4 (2013): 496-502.

[5] Yakimov, Pavel Yurievich. "Tracking traffic signs in video sequences based on a vehicle velocity." *Computer optics* 39.5 (2015): 795-800

[6] Stallkamp, Johannes, et al. "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition." *Neural networks* 32 (2012): 323-332

[7] Houben, Sebastian, et al. "Detection of traffic signs in real- world images: The German Traffic Sign Detection Benchmark." The 2013 international joint conference on neural networks (IJCNN). Ieee, 2013.

[8] V.Terehov, D. Efimov, I. Tiukin, Neural network control system [in Russian], Textbook for high schools, High school, 2002.

[9] Clevert, Djork-Arné, Thomas Unterthiner, and Sepp Hochreiter. "Fast and accurate deep network learning by exponential linear units (elus)." arXiv preprint arXiv:1511.07289 (2015).

[10] Shustanov, Alexander, and Pavel Yakimov. "CNN design for real-time traffic sign recognition." Procedia engineering 201 (2017): 718-725.

[11] Huval, Brody, et al. "An empirical evaluation of deep learning on highway driving." arXiv preprint arXiv:1504.01716 (2015).

[12] Abadi, Martín, et al. "Tensorflow: A system for large-scale machine learning." 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16). 2016.

[13] Jin, Junqi, Kun Fu, and Changshui Zhang. "Traffic sign recognition with hinge loss trained convolutional neural networks." IEEE Transactions on Intelligent Transportation Systems 15.5 (2014): 1991-2000.

[14] Sermanet, Pierre, and Yann LeCun. "Traffic sign recognition with multi-scale convolutional networks." The 2011 International Joint Conference on Neural Networks. IEEE, 2011.

[15] Qian, Rongqiang, et al. "Robust Chinese traffic sign detection and recognition with deep convolutional neural network." 2015 11th International Conference on Natural Computation (ICNC). IEEE, 2015.

[16] Mathias, Markus, et al. "Traffic sign recognition—How far are we from the solution?" The 2013 international joint conference on Neural networks (IJCNN). IEEE, 2013.