Traffic Sign Recognition using CuDNN

Aditya Das
UG Scholar, CSE Department, SRM Institute of Science and Technology, Vadapalani, Chennai, Tamilnadu, India.
adityadas_a@srmuniv.edu.in

Jishnu De Sarkar
UG Scholar, CSE Department, SRM Institute of Science and Technology, Vadapalani, Chennai, Tamilnadu, India
jishnu_r@srmuniv.edu.in

Arka Dasgupta
UG Scholar, CSE Department, SRM Institute of Science and Technology, Vadapalani, Chennai, Tamilnadu, India
arkadasgupta_m@srmuniv.edu.in

Done under the supervision of:
Mrs. R. Logeshwari B.E., M.E.
Assistant Professor,
Computer Science Engineering, *SRM Institute of Science and Technology, Vadapalani, Chennai, Tamilnadu, India*

***Abstract* – Object detection has been a prime area of focus for researchers across the world for several years now. With newer technology being developed at a very rapid rate, scientists have forever been on a quest to develop models with highest accuracy. This paper aims at providing a novel approach to the problem of detecting objects and comparing it with the existing methods. It has been observed that CuDNN (NVIDIA CUDA® Deep Neural Network library) can leverage the potential of GPU (Graphical Processor Units) over conventional CPU cores and can hence achieve significantly higher accuracy than other traditional neural network models. We use CLAHE (Contrast Limited Adaptive Histogram Equalization) for the preprocessing of the data and then take the help of CuDNN library to enhance the results even further. Moreover, computations done using CuDNN have been observed to run notably faster.**

I. Introduction

With automation penetrating into each and every industry, image recognition in the automotive industry has seen some of the most innovative solutions to make people's lives easier. TSR (Traffic Sign Recognition) is a technology which automatically recognizes traffic signs along the road, including speed limit signs, yield signs, merge signs, etc. The ability to automatically recognize traffic signs enables us to build "smarter cars". TSR technology is used in a wide range of applications from driver-assistance systems like a fully autonomous vehicle to embedding it into a dashboard camera system.

The two prime components in a TSR system are the following: the detection component and the classification component. Over the years, some noteworthy work has been observed in the latter, however, the former still remains as an area with plenty of scope for improvement. The reason behind the same being, detection requires much more complex methods. The complexity of detection comes into the picture because the detection process requires two steps to achieve accurate localization of objects. Firstly, candidates that only provide rough localization of objects must be obtained from numerous proposals. Secondly, coarse localization must be refined to achieve precise localization.

Recently a significant amount of progress has been made in detecting traffic sign signals due to the advancements in the deep convolutional neural networks. As a result of which, researchers have found it difficult to decide which architecture and configuration are best suited to tasks such as feature extractors. (e.g. Inception, Residual Network) and detection architectures (e.g. Faster R-CNN and SSD).

It is always advised to design an algorithm to make performance on specific tasks better in comparison to the ability to be applied in other scenarios. Since, humans easily detect road signs due to their distinct shapes and high saturated colours as well as the fact that the signs are usually very easy to distinguish from the background it seems fair to assume that algorithms would be able to use the same characteristics to differentiate between the symbols and their backgrounds. However, in reality due to the difference in size of the symbols in comparison to their background it's a seemingly difficult task for an system to identify or detect an object with good accuracy.

The prime objective of the following paper is 1) compare and contrast the performance while running image recognition algorithms while using NVIDIA® CUDA cores compared to regular CPU cores offered by Intel® and other processor manufacturers 2) the paper sheds light on the advantages and disadvantages of each approach and 3) this paper demonstrates that CUDA based processing can achieve better performance compared to alternatives while implementing DNN architecture.

This paper is divided into four sections. Section I, aims at highlighting the purpose and objective of the paper at a high level. Section II, gives an idea about the work that has already been done in the following domain. Section III, aims at explaining the approach in detail starting from capturing the image to applying the CLAHE algorithm to enhance the captured image. Section III explains the observations from the conducted experiments and section IV shows some of the areas where the following study can be implemented and draws the conclusion respectively.

II. RELATED WORKS

A lot of work has been done on the topic of image processing and sign recognition before. The fundamental difference between the previous approaches and ours is: 1) in the pre-processing phase previous approaches mark the detected signs with yellow boxes [3] or otherwise simply crop the image from the overall input image [1] [2]. However we use CLAHE, which increases the contrast and brightness of the image making it easier for the classifier. 2) Instead of using regular CPU cores [1] [2] [3] as done by previous approaches [4] [5] [8] we use NVIDIA® CUDA cores, helping us to bring down the time taken for the model to process an image upto 66% more.

III. DETECTION OF IMAGES

Before we go into the details of how the process works, let’s get a quick introduction about the dataset that we will be using in this paper. The dataset we’ll be using to teach our own custom traffic sign classifier is the German Traffic Sign Recognition Benchmark (GTSRB). . The GTSRB dataset consists of 43 traffic sign classes and nearly 50,000 images.

Now that we've established the dataset that we are becoming to use, let’s go over a few of the challenges that we encountered using this particular dataset. the first being that images are low resolution, and second, have poor contrast. These images are pixelated, and in some cases, it’s extremely challenging, if not impossible, for the human eye and brain to acknowledge the sign. So to overcome this challenge we took the help of CLAHE (Contrast Limited Adaptive Histogram Equalization). Adaptive histogram equalization (AHE) could also be a computer image processing technique used to improve contrast in images. It differs from ordinary histogram equalization within the respect that the adaptive method computes several histograms, each sort of a definite section of the image, and uses them to redistribute the lightness values of the image. it's therefore suitable for improving the local contrast and enhancing the definitions of edges in each region of an image

In the real-world, traffic sign recognition happens to be a two-stage process. First localization, where we detect and localize where in an input image/frame a traffic sign is. Next comes recognition, where we take the localized ROI and recognize and classify the traffic sign. Deep learning object detectors can perform localization and recognition during a single forward-pass of the network.

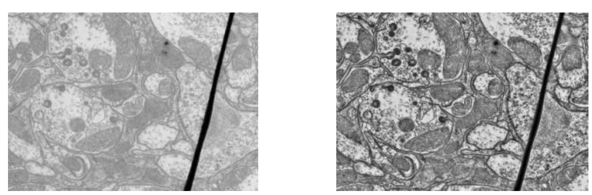
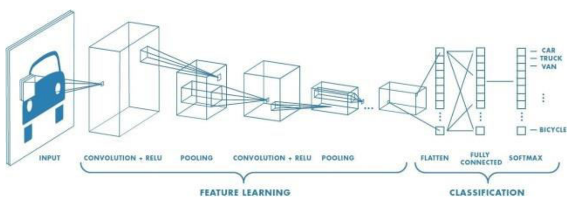


Fig. 1. Original Image (left), Image processed by CLAHE (Right)

We saw a significant increase in the accuracy of our model when we employed CLAHE preprocessing architecture into our model. After using CLAHE the accuracy of the image detection got an increase of upto 5 percent. Ordinary histogram equalization uses an equivalent transformation derived from the image histogram to rework all pixels. This works well when the distribution of pixel values is analogous throughout the image. However, when the image contains regions that are significantly lighter or darker than most of the image, the contrast in those regions won't be sufficiently enhanced.

Now that we have an idea about the preprocessing algorithm, let's take a look at the steps to finally process the image. The process starts with an input image, for example that of a vehicle. Computers see an input image as an array of pixels and size of the array depends on the image resolution. In order for deep learning CNN models to train and test, each input image will be passed through a series of convolution layers with filters (Kernels), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. The below figure is a complete flow of CNN to process an



input image and classifies the objects based on values.

Fig. 2. Neural network with many convolutional layers

Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel. Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying filters. Stride is the number of pixels shifted over the input matrix. Sometimes, filters do not perfectly fit the input image. We have two options: Pad the picture with zeros (zero-padding) so that it fits or Drop the part of the image where the filter did not fit. This is called valid padding which keeps only the valid part of the image.

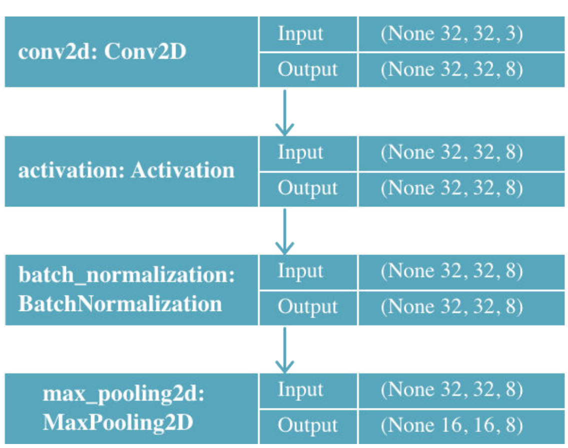


Fig. 3. Step by step demonstration of the chosen CNN model.

ReLU stands for Rectified Linear Unit for a non-linear operation. The output is $f(x) = \max(0,x)$. ReLU's introduces non-linearity in our ConvNet. Since, the real world data would want our ConvNet to learn would be non-negative linear values. There are other nonlinear functions such as tanh or sigmoid that can also be used instead of ReLU. Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling is also called subsampling or downsampling which reduces the dimensionality of each map but retains important information. The layer we call the FC layer, we flatten our matrix into vectors and feed it into a fully connected layer like a neural network.

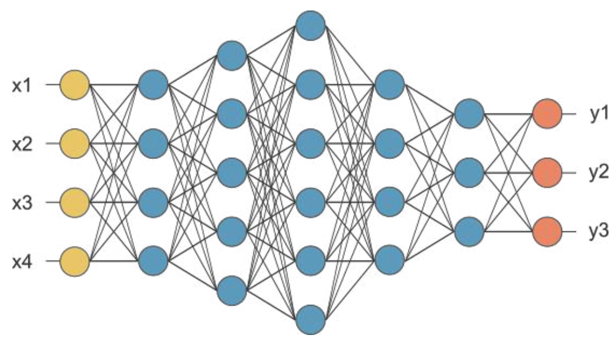


Fig. 4. After pooling layer, flattened as FC layer.

Let's now look at how using a GPU can bring a fundamental difference in the accuracy of an image recognition engine, when compared to a traditional CPU. A CPU (central processing unit) is often called the "brain" of the computer. It is required to run the majority of engineering and office software. However, there are a multitude of tasks that can overwhelm a computer's central processor. That is when using GPU becomes essential for computing. A GPU (graphics processing unit) is a specialized type of microprocessor, primarily designed for quick image rendering. GPUs appeared as a response to graphically intense applications that put a burden on the CPU and degraded computer performance. GPUs can accurately process millions of images to find differences and similarities. Hence, using a GPU for the task of recognising the road signs makes a lot of sense since the entire process is very uniform in nature.

III. OBSERVATIONS FROM THE EXPERIMENT

After conducting the following set of experiments we were able to provide some substantial evidence that using GPU instead of a CPU has a significant impact on the time required to process an image. It was noted that using GPU decreased the time from 0.3 sec/image to just 0.1 sec/image which is an efficiency rate increase of upto 66%.

In case of using CLAHE for pre-processing the input images, we were able to push the accuracy of the existing model [1] from ≈ 92% to ≈ 96%. Which is a 4% increase in the overall accuracy of the model.

IV. CONCLUSIONS AND FUTURE SCOPE

From the results of the experiment it can be conducted that using a GPU is a much better alternative over using a CPU, and can bring down the amount of time required to process the images by a huge margin. In addition to this, it can also be concluded that, better the preprocessing algorithm, better the results. Hence using better preprocessing algorithms such as CLAHE can significantly impact the accuracy of the model as most of the images then get correctly classified.

Before we proceed to the list of areas that our project can be applied to, let's take a look at certain technical limitations in our project. First, instead of using a classic CNN algorithm we can use architectures such as R-CNN, RFCN and FCN. If we integrate computer vision, we can apply the CNN algorithm to a particular part of the image, enhancing the accuracy of the model even further. In addition to CLAHE for pre-processing images, we can apply filters such as Wiener filter. Wiener filter helps in reducing the overall noise of the image along compensating for motion blur caused during capturing the image.

Recognition of road signs can be applied in multiple applications for example: It can be integrated into autonomous vehicles. It can be used for real time identification of the speed limits in modern cars, where the speed limit of a road is directly shown on their dashboards. Even in case of regular dashboard cameras, having a talkback feature can directly read the signs for the driver without causing any interruptions. It can also be made into an app which can be used by foreigners to understand the road signs of a different country.

V. REFERENCES

[1] Qing Tang, Kang-Hyun Jo. "Analysis of Various Traffic Sign Detectors Based on Deep Convolutional Network." *Intelligent Transportation Systems Conference 2006. ITSC '06. IEEE, pp. 811-816, 2006.*

[2] Shaung Xu, Deqing Niu, Bo Tao, Gongfa Li. "Convolutional Neural Network Based Traffic Sign Recognition System." *Biomedical Circuits and Systems Conference (BioCAS) 2014 IEEE, pp. 544-547, 2014.*

[3] A. W. Setiawan, T. R. Mengko, O. S. Santoso and A. B. Suksmono, "Color retinal image enhancement using CLAHE," *International Conference on ICT for Smart Society*, Jakarta, 2013, pp. 1-3.

[4] S. Jung, U. Lee, J. Jung and D. H. Shim, "Real-time Traffic Sign Recognition system with deep convolutional neural network," *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Xi'an, 2016, pp. 31-34.

[5] A.Krizhevsky, I.Sutskever, and G.Hinton, "Image Net classification with deep convolutional neural networks". *In NIPS, 2012*.

[6] P. Sermanet and Y. Lecun, "Traffic sign recognition with multi-scale convolutional networks." *In Proc. IJCNN, 2011.*

[7] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner, and W. vonseelen, "An image processing system for driver assistance," in *Proc. Ieee Int. Conf. Intell. Veh., 1998*, pp.481-486.

[8] Malik, Z., and Siddiqi, I., "Detection and Recognition of Traffic Signs from Road Scene Images." *International Conference on Frontiers of Information Technology IEEE, 2015:330-335.*

[9] Wahyono, and Jo, K., H., "A comparative study of classification methods for traffic signs recognition." *IEEE International Conference on Industrial Technology IEEE*, 2014:614-619.

[10] Le, Guoqing, et al. "Traffic sign recognition based on PCANet." *Advanced Information Management, Communicates, Electronic and Automation Control Conference IEEE*, 2017:807-811

[11] R. A. Hummel, "Image Enhancement by Histogram Transformation". *Computer Graphics and Image Processing 6*.

[12] Thamizharasi Ayyavoo, Jayasudha John "Illumination pre-processing method for face recognition using 2D DWT and CLAHE " *in SuseelaIET Biometrics 7.*

[13] M.A. Garcia-Garrido, M.A. Sotelo, E. Martin-Gorostiza, "Fast traffic sign detection and recognition under changing lighting conditions", Intelligent Transportation Systems Conference 2013. ITSC '13. IEEE, pp. 629-634, 2013c=