



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** XI **Month of publication:** November 2023

DOI: <https://doi.org/10.22214/ijraset.2023.56472>

www.ijraset.com

Call: ☎ 08813907089

E-mail ID: ijraset@gmail.com

Traffic Sign Classification Using ASIC

Asst. Prof. G. Krishna Kishore¹, A. Shivarani², M. Priyanka³, P. Madhavi⁴, B. Sandhya⁵

ECE Dept, G. Narayanamma Institute of Technology and Science, Hyderabad, India

Abstract: *The main aim of this project is to focus on traffic signs classification using Tensor flow and deploying the model onto an ASIC for engineering applications. The classification is done using Convolutional Neural Networks (CNNs) in TensorFlow, where the model is trained using a large dataset of traffic sign images. The training images are pre-processed and labelled, and the model is fine-tuned using transfer learning techniques. After the model is trained, it is tested with a separate dataset to ensure its accuracy. Once the model is finalized, it is optimized for deployment onto an ASIC using Micro Python. The ASIC is a custom-designed hardware that is optimized for the specific task of traffic sign classification. The performance of the ASIC is then compared with a standard computer to evaluate its speed and efficiency. This project demonstrates the ability to implement machine learning algorithms onto custom-designed hardware for specific engineering applications, specifically for the purpose of traffic sign classification.*

Keywords: CNN, ASIC, Speed limit, Zebra crossing, Car/bus lane, Mobile Net V2

I. INTRODUCTION

Traffic sign classification plays a pivotal role within the realm of intelligent transportation systems, contributing significantly to road safety and providing essential driver assistance. Their purpose is to regulate and guide road users, ensuring compliance with established road rules. In our increasingly automated world, we have come to rely on the simplification of tasks. However, while driving, our attention often remains fixed on the road, causing us to overlook crucial signs on the roadside. This not only poses a danger to the driver but also to others on the road. To address this issue, it is essential to alert drivers without requiring them to divert their attention.

Traffic sign detection and classification serve as key components in achieving this objective. By detecting and recognizing traffic signs, a system can proactively alert the driver about approaching signs, fostering a safer driving environment. This approach not only enhances road safety but also alleviates the driver's stress when navigating unfamiliar or complex roads, as it eliminates the need to decipher signs.

The primary aim of this research is to develop a Convolutional Neural Network (CNN)- based model that can educate people about traffic signs, an aspect of daily life often overlooked yet of utmost importance. The objective is to achieve the highest level of accuracy possible, ensuring that the proposed model can be adopted with confidence by anyone.

II. BASIC PRELIMINARIES AND RELATED WORK

Traffic sign classification requires an appropriate dataset, powerful and efficient embedded artificial intelligence (AI) chip kendryte k210 with camera and SD card slot.

- 1) **Data collection:** Gather a dataset of traffic sign images. This dataset should cover a wide range of traffic signs, including various shapes, colours, and symbols. The dataset should also include diverse lighting conditions and angles.
- 2) **Data preprocessing:** Preprocess the collected dataset to ensure consistent image sizes, colours, and orientations. The data preprocessing steps involve resizing, normalization, and the implementation of data augmentation techniques, including rotation, translation, and flipping.
- 3) **Model training:** Train a deep learning model on the pre-processed dataset, including traffic sign classification using popular deep learning frameworks like TensorFlow to build and train the model.
- 4) **Model optimization:** Once you have trained your model, optimize it for deployment on the Kendryte K210. This may involve techniques such as model quantization or pruning which reduces the model size and the computation. Model conversion: Convert the trained model to a format compatible with the Kendryte K210. The Kendryte K210 supports the TensorFlow Lite Micro framework, so you may need to convert your model to the TensorFlow Lite format.
- 5) **Deployment on Kendryte K210:** Finally, deploy the optimized and converted model onto the Kendryte K210. Utilize the KPU on the Kendryte K210 to perform real-time inference on traffic sign images.

III. PROPOSED WORK

A. Dataset

A dataset is required for the classification. The model must be trained with the dataset which consists of images. The foundation for starting a detection or classification task is the collecting of a large dataset. It is necessary to use a large and high-quality dataset when training a model. The training set and testing set are the two main divisions of this dataset. The information used to create and train the model is contained in the training dataset.



Fig.1 Dataset of zebra crossing, speed limit 50, car/bus lane

B. Transfer Learning Techniques

Transfer learning is essentially the process of training and making predictions on a new dataset using a model that has already been trained on one dataset. "A pre-trained model is a saved network that was previously trained on a large dataset, typically on a large-scale image classification task." To categorize the photos of traffic signs in this project, we will apply transfer learning. Learning transfer using mobile net version 2 The Google-developed Mobile Net V2 model was pre-trained on the ImageNet dataset, which contains 1.4 million photos and 1000 classes of web images.

C. The CNN Architecture

The main stages of the convolutional process and these are the extra layers added to the already trained model:

- 1) *Convolution*: A localized level of convolution is a technique used to extract object attributes from a picture. As a result, the network can distinguish certain patterns within the image and within the full image. A computer scans a sector of the image (often a 3x3 section) and multiplies it by a filter as part of this process, which involves element-wise multiplication. Feature maps are created as a result of this.
- 2) *Nonlinearity (ReLU)*: After convolution, nonlinearity is added by applying an activation function like ReLU. CNNs frequently use the ReLU activation function, which converts negative pixel values to zero.
- 3) *Pooling process*: In order to reduce the dimensionality of the input photos, the pooling process is used to combine the images. This process is crucial for reducing the operation's computing complexity. The network computes fewer weights when the dimensionality is reduced, minimizing the likelihood of overfitting. Setting parameters like stride and size is typical when pooling.
- 4) *Fully Connected Layers*: Building a conventional CNN needs fully connected layers as the last phase. Here, neurons from every layer before this are connected to the layer after. The final prediction is produced by classifying the signs present in the input image using a SoftMax activation function.
- 5) *Data Preprocessing*: The configuration of the output layers will be manually adjusted to suit the specific requirements of our problem. We will add the final layers to the mobile Net. The preprocessing steps involved in this process include resizing the image, converting it to a NumPy array, and applying preprocessing specific to the Mobile Net model.

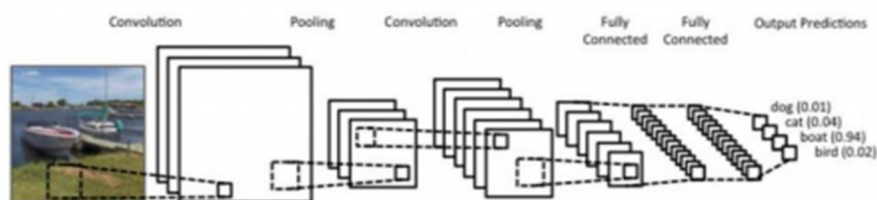


Fig.2 CNN architecture

D. Training And Testing

To train our model, we will utilize the available image data. The training process will consist of five epochs, with each epoch comprising 24 steps per epoch. These steps per epoch signify the number of weight updates made within each training cycle. It's worth noting that an ideal value for steps per epoch aligns with the number of samples per batch. We will then proceed to fit the training data generators and input the specified parameters into the model, initiating the training process. Once we ensure the model is ready to classify the images, the model is been tested with the test dataset. If it identifies with good accuracy the further step is model conversion.

```
Epoch 1/5
24/24 [=====] - 21s 567ms/step - loss: 0.4565 - accuracy: 0.8148
Epoch 2/5
24/24 [=====] - 12s 493ms/step - loss: 0.1394 - accuracy: 0.9597
Epoch 3/5
24/24 [=====] - 11s 460ms/step - loss: 0.0751 - accuracy: 0.9785
Epoch 4/5
24/24 [=====] - 11s 469ms/step - loss: 0.0525 - accuracy: 0.9866
Epoch 5/5
24/24 [=====] - 11s 459ms/step - loss: 0.0395 - accuracy: 0.9857
```

Fig 3. Number of epochs and step_per_epoch

E. Model Conversion

After the model has been trained and tested, optimize it for deployment on to the Kendryte K210. In order to reduce the model size and computational requirements, approaches like model quantization or pruning may be used. Transform the training model into a Kendryte K210-compatible format for that it needs to be converted to the TensorFlow Lite format because the Kendryte K210 supports the TensorFlow Lite Micro framework. The TensorFlow lite mode is then converted to K-Model. The model is dumped inside SD card. The path of the K-Model is given for deployment of the optimized and converted model onto the Kendryte K210 to complete the process. Where kendryte K210 consists of a camera, it can capture the images, detects and classifies the traffic signs.



Fig 4. Zebra Crossing



Fig 5 Speed Limit 50



Fig 6 Car/Bus lane

IV. RESULTS AND DISCUSSION

A Convolutional Neural Network (CNN) architecture-based enhanced traffic sign classification system has been successfully developed on an Application-Specific Integrated Circuit (ASIC). This state-of-the-art model was created with the intention of identifying and categorizing traffic signs, accepting user feedback, and assessing its accuracy through extensive testing. A remarkable accuracy of 97.74% was attained after extensive training, which involved iteratively running the entire dataset through the neural network for 24 full forward and backward propagations (epochs). This accuracy outperformed all previously used baseline models.

V. CONCLUSION

When used to categorize three different traffic signs, the suggested model, which is based on Convolutional Neural Networks (CNN), produced an accuracy rate of 97.74% and 98.93%. It was painstakingly created and put to a dataset that had been carefully selected. This exceptional level of accuracy highlights the strength and potency of the suggested model. Its high level of accuracy in correctly detecting traffic signs is crucial in reducing the likelihood of accidents and, as a result, improving road safety. This approach is a huge advance in the direction of reducing accidents and improving all aspects of road safety. Future work will examine the model's performance using a variety of traffic sign datasets from other nations, address potential memory storage issues with AI chips, and increase the dataset by including more traffic sign images. These steps promise to enhance the model's capabilities and its potential to make an even more significant impact on road safety in the future.

REFERENCES

- [1] D. Bodington, E. Greenstein, and M. Hu. Implementing machine learning algorithms on gpus for real-time traffic sign classification. 2015
- [2] F. Boi and L. Gagliardini. A support vector machines network for traffic sign recognition. The 2011 International Joint Conference on Neural Networks, pages 2210–2216, July 2011.
- [3] Z. Chen, X. Huang, Z. Ni, and H. He. A gpu-based real-time traffic sign detection and recognition system. 2014 IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems (CIVTS)
- [4] D. Ciregan, U. Meier, and J. Schmid Huber. Multi-column deep neural networks for image classification. 2012 IEEE Conference on Computer Vision and Pattern Recognition, June 2012.
- [5] L. Davis, "Visual navigation at the University of Maryland," in Proc. Int. Conf. Intelligent Autonomous Systems 2, Amsterdam, The Netherlands, 1989.
- [6] EDickmans, "Machine perception exploiting high level spatiotemporal models," presented at the AGARD Lecture Series 185, Madrid, Spain, Sept.
- [7] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4651–4659, 2016.
- [8] K. H. Lim, K. P. Seng, and L.-M. Ang, "Mamo Lyapunov theory based rbf neural classifier for traffic sign recognition," Applied Computational Intelligence and Soft Computing, vol. 2012, p. 1, 2012.
- [9] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun, "Deep image: Scaling up image recognition," arXiv preprint arXiv:1501.02876, 2015.
- [10] X. B. Quan and W. X. Xiong, "Real-time embedded traffic sign recognition using efficient convolutional neural network," IEEE Access, 2019.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)