Recognition of Traffic Signs Using Machine Learning

R.Sai chandra[1], M.Vinay[2], Y.Pavan [3], K.Naveen[4] and D.Siva Raja Kumar[5*]

[1234]Student, Department of CSE(AI&ML),CMR Engineering College,Hyderabad,Telangana

[5] Professor, Department of CSE(AI&ML),CMR Engineering College,Hyderabad,Telangana

ABSTRACT:

Hey there! Nowadays, Convolutional Neural Networks (CNN) are really stepping up to solve object recognition tasks. These networks are super effective and speedy, making them perfect for all sorts of computer vision missions. In this article, we suggest a way to recognize traffic signs using a CNN. The paper showcases different CNN setups, comparing how they work. We trained the neural network with TensorFlow and CUDA for fast processing. The whole traffic sign detection process happens in real-time on a mobile GPU. Our tests show that our computer vision system works really well!

1.INTRODUCTION

Traffic sign recognition is a big challenge in academics and industry. These systems are crucial for the future of Artificial Intelligence (AI) and are for Advance Driver Assistance Systems (ADAS). Recogn traffic signs shows us that signs can come in different shapes, colors, and graphics, helping drivers anticipate what they might encounter on the road. Sometimes signs aren't easy to see, leading to confusion or even accidents. In some cases, signs can be hard to recognize due to various conditions, making it tough for both humans and machines.

To tackle these challenges, we can use generated data to make it easier for classifiers. Our project focuses on recognizing signs through classification and extraction. There are many ways to identify a sign, but their appearance stays consistent. We aim for a detection accuracy above 90% to match the real-world scenario using CNN technology.

Friendly vibes are shared when the system spots a sign - not exactly like the real one but close enough to keep everyone safe on the road.

2.RELATED WORKS

Author[1] tell us about this cool way to find and follow road signs in busy traffic. First, we use two fancy neural networks to spot color and shape on the signs in pictures of the roads. These images are how we find where the signs are. It's all about fuzzy-set stuff, you know? Then comes the tracking part. We keep an eye on the signs we found as they move through different pictures using a Kalman filter. The tests show that our method works great at spotting and keeping up with road signs, even when it's super messy outside. Cool, right?

Author[2] has come up with a way spot and understand road signs better by using a two-step process. First off, we look for the signs by checking their shape, corners, and where they appear in a messy picture. Then, we figure out what the signs are trying to tell us by using fancy math stuff like neural networks and clever algorithms. Our tests show that our method works pretty well, correctly spotting signs about 95.5% of the time! Plus, it's tough enough to handle tricky situations on the road. This idea could really help make driving and more relaxing for everyone out there.

Author[3] tried using Convolutional Networks (ConvNets) for sorting traffic signs during the GTS competition. ConvNets are super cool structures inspired by biology that learn features automatically. While some popular vision methods rely on hand-crafted features like HOG or SIFT, ConvNets learn features at every stage from data that match the task. We tweaked the classic ConvNet design by giving 1st stage features along with 2nd stage features to the classifier. Our setup achieved a really great accuracy of 98.97% in phase I of the competition (the top-ranking entry got 98.98%), beating human performance at 98.81%, all with color images sized 32×32.

Later on, we set a new record of 99.17% in experiments post-phase I by bumping up network capacity and switching to greyscale images instead of color. Surprisingly, even random features gave pretty good results at 97.33%.

3.METHODOLOGY

3.1 Dataset Acquisition:

- Collect a comprehensive dataset containing diverse images of traffic signs. Ensure that the dataset covers various types of signs, lighting conditions, and potential scenarios encountered on roads.
- Annotate the dataset with corresponding labels for each traffic sign class.

3.2 Preprocessing:

- Resize all images to a consistent dimension suitable for model input.
- Normalize pixel values to a standard range (e.g., 0 to 1).
- Augment the dataset with techniques like rotation, flipping, and brightness adjustments to enhance model generalization.

3.3 Building CNN Model:

- Design a Convolutional Neural Network (CNN) architecture suitable for image classification. Consider using popular frameworks like TensorFlow or PyTorch.
- Include convolutional layers for feature extraction and dense layers for classification.
- Incorporate pooling layers to reduce spatial dimensions and improve computational efficiency.

3.4 Training Model:

- Split the dataset into training and validation sets.
- Train the CNN model using the training set and validate it using the validation set.
- Utilize a suitable loss function (e.g., categorical cross-entropy) and an optimizer (e.g., Adam) during training**.**

3.5 Testing Mode

- Evaluate the trained model on a separate testing dataset to assess its generalization to unseen data.
- Measure performance metrics, including accuracy, precision, recall, and F1 score, to gauge the model's effectiveness.
- Analyze model predictions on individual traffic sign images to identify areas for improvement.

3.6 Voice Alert Integration:

- Develop a system for integrating voice alerts into the recognition process.
- Associate specific voice alerts with each recognized traffic sign class.
- Implement real-time communication between the model predictions and the voice alert system.

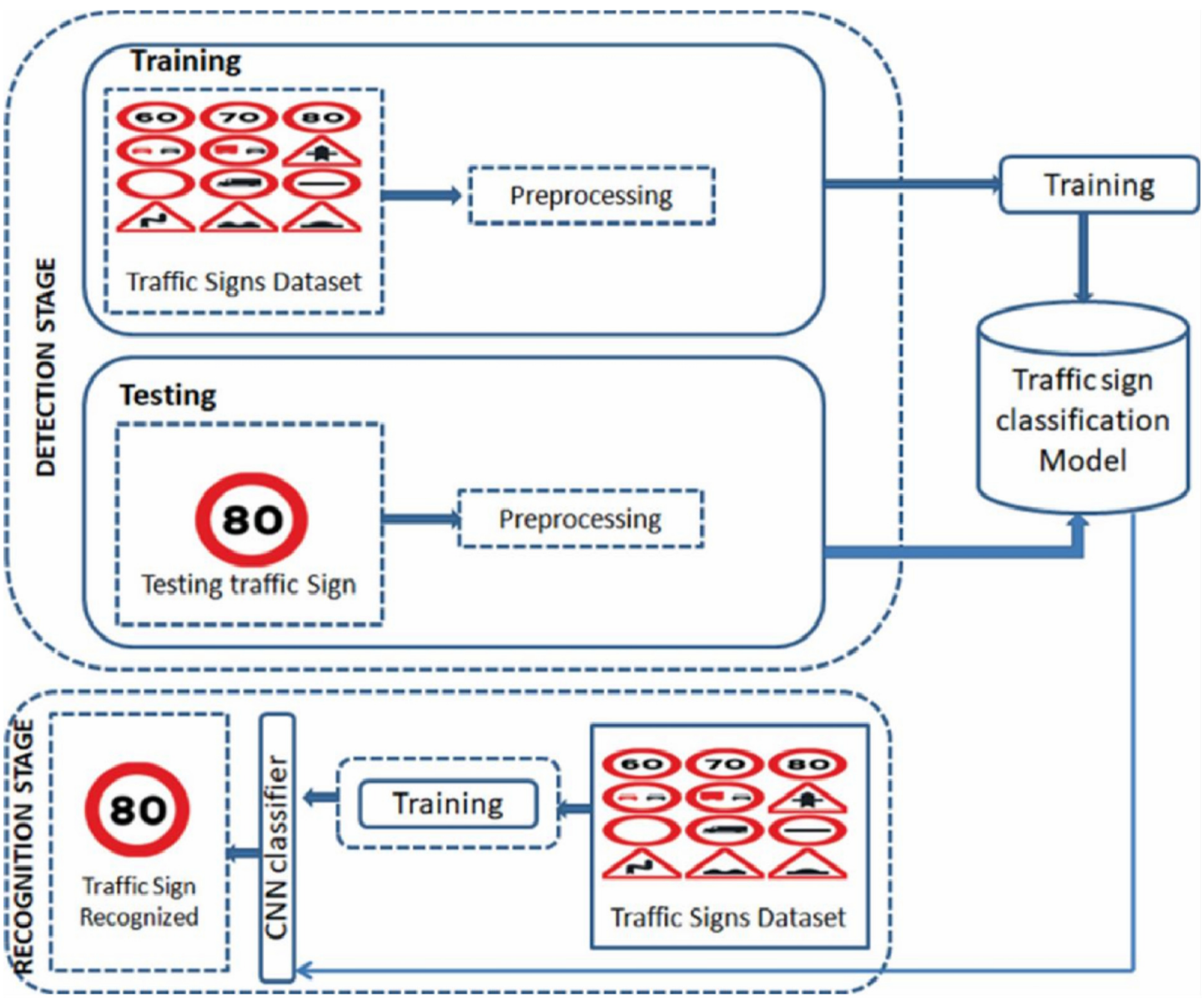
3.7 System Integration and Deployment:

- Integrate the trained CNN model and voice alert system into a cohesive application.
- Develop a user interface for visualizing real-time traffic sign recognition and corresponding voice alerts.

ISSN: 10123407

4. SYSTEM DESIGN

System Architecture is a graphical representation of steps. It shows steps in sequential order and is widely used in presenting the flow of algorithms, workflow or processes. Typically, a flowchart shows the steps as boxes of various kinds, and their order by connecting them with arrows.



5. RESULT

6.CONCLUSION

Hey there! Let's wrap things up, shall we?

This paper talks about using a classification to recognize traffic signs.

By adding in some preprocessing and localization steps, the new method for classifying traffic signs is doing really well - it's getting99.94% of images right!

To make this happen, we used the TensorFlow framework.

Our traffic sign recognition algorithms can handle real-time video streams with high resolution, from further away and with better quality than other systems. The FullHD resolution means we can spot a traffic sign up to 50 meters away.

We put this method into action on device with an Nvidia Tegra K1 processor and used CUDA to speed things up.

In the future, we want to train the CNN to recognize more kinds of traffic signs and handle different weather conditions. Plus, we hope to use it for object detection too.

7.FUTURE SCOPE

Real-time Recognition: Optimize model and implementation to perform real-time recognition of traffic signs. This could involve reducing inference time, potentially through model compression techniques or hardware acceleration.

Multi-Language Support: Expand the system to recognize and provide information about traffic signs in multiple languages. This would be particularly useful in multicultural or multilingual regions.

Improved Accuracy and Robustness: Continuously train and fine-tune your model with more diverse datasets to improve its accuracy and robustness. This could involve collecting data from different environmental conditions, such as different times of day, weather conditions, and camera angles.

Integration with Navigation Systems: Integrate your traffic sign recognition system with navigation systems or GPS applications to provide enhanced driving assistance. This could include providing alerts or route recommendations based on detected traffic signs.

Semantic Segmentation: Instead of just recognizing the presence of a traffic sign, consider implementing semantic segmentation to precisely locate and classify different elements of the traffic scene. This could help in better understanding the context and improving the system's response.

Vehicle-to-Infrastructure Communication: Explore possibilities for integrating your system with vehicle-to-infrastructure (V2I) communication technologies. This could enable your system to receive information about traffic signs from roadside infrastructure, further enhancing its capabilities.

User Interface Improvements: Enhance the user interface for both text and voice outputs to make them more user-friendly and intuitive. Consider adding features such as natural language processing for voice interactions or a more visually appealing display for text output.

Adaptive Learning: Implement adaptive learning techniques to allow the system to improve over time based on user feedback and new data. This could involve incorporating reinforcement learning or online learning strategies.

Edge Computing: Investigate the feasibility of deploying your system on edge devices, such as onboard vehicle computers or mobile devices, to reduce latency and dependency on internet connectivity.

Security and Privacy Considerations: Pay attention to security and privacy considerations, especially if your system is deployed in real-world scenarios. Implement encryption and data anonymization techniques to protect user data and ensure the system's integrity.

8.REFERENCES

[1] C.Fang, S. Chen, C. Fuh, "Road-sign detection and tracking", IEEE Transactions on Vehicular Technology, Vol. 52 pp. 1329–1341, Issue 5,2021.

[2]W-J. Kuo, C.-C. Lin, "Two-stageroad sign detection and recognition", IEEE International Conference on Multimedia and Expo, Vol.3 pp. 1427–1430, Issue 2018.

[3]Pierre Sermanet, Yann LeCun, "Traffic Sign Recognition with Multi-Scale Convolutional Networks", Proceedings of International Joint Conference on Neural Networks, San Jose, California, USA, Issue July 31 – August 5, 2019.