CUSTOM CNN BASED TRAFFIC SIGN BOARD RECOGNITION AND VOICE ALERT SYSTEM

[1]Dr. G. Kalpana, [2]Nune S V S D L Pravallika, [3]Yalavarthi Bhanu Lekha,[4]Nagolu Ashwini , [5]Kistapuram Keerthana

[1]Associate Professor & HOD, [2,3,4,5]UG Students, Dept. of CSE (AI & ML), Malla Reddy Engineering College for Women (Autonomous), Hyderabad, India.E-mail: dr.gkalpanamrecw@gmail.com

ABSTRACT

In the field of artificial intelligence and technical innovation, numerous researchers and significant companies like Tesla, Uber, Google, Mercedes-Benz, Toyota, Ford, and Audi are collaborating on autonomous vehicles and self-driving cars. Hence, the vehicles' ability to read traffic signs and respond appropriately is necessary for such technology to be effective. You must obey various traffic indicators, including traffic lights, turn signals, speed limits, exit and entry limitations, pedestrian crossings, and others, to be able to drive safely. In a similar way autonomous vehicles must understand these signals and make decisions so as to achieve precision.Traffic signs categorization is a way for identifying the class to which a traffic sign belongs. It is useful in conventional automobiles even though it is not an autonomous one. For there to be a secure and predictable flow of traffic, there must be road signs. Auto accidents are greatly influenced by negligence in the reading and interpretation of traffic signs. The proposed device helps drivers recognise traffic signs and alerts them via speaker so they can take the proper decisions. The recommended strategy helps identify and classify photos of traffic signs by employing a specially trained convolutional neural network. A set of classes is built and trained in order to improve the precision of a particular set of data.We used the German Traffic Sign Benchmarks Dataset that includes 39,209 photos of traffic signs in about 43 categories. About 98.52 percent of the execution was accurate. A voice alert is broadcast throughout the speaker after the system recognises the sign to inform the driver. The system's goal is to protect the driver, passengers, and pedestrians from harm. Using a Custom Convolutional Neural Network, Keras, and other built-in Python modules, we developed a model for the classification of traffic signs seen in the image into numerous grouping for this Deep Learning project.

INTRODUCTION

The vehicle-aided driving system has rationalized the preceding driving mode in current years by means of the rise of Artificial Intelligence (AI). The creation of autonomous vehicles necessitates not only auxiliary driving systems but also quick and precise traffic sign detection.The TSR approaches mostly rely on visual data, such the design and color of traffic signals. However, in real-time tests, the typical TSR algorithms have problems, such as life formsimply constrained by road circumstances, such as lighting, camera angle, blockage, vehicle speed, and so forth.Artificial neural network limitations have been significantly reduced thanks to ongoing advancements in computer technology, ushering in a golden age for machine learning research and development. When processing information, a deep neural network model replicates the neuronal architecture of the human brain. The sturdiness and simplification of the algorithms could be enhanced by means of this neural network model rather of the traditional TSR models to take out the useful characteristics as of the road image.

Traffic sign's location and recognition is a critical viewpoint for benevolentsafety to all street clients. Traffic Sign Recognition and grouping can be utilized to consequently recognize traffic signs. This is done consequently by the framework as the traffic sign is identified and the sign name(text message) is shown and also alerts the driver with a voice message. Thus regardless of whether any sign is missed by the driver or has any slip-by in fixation, it will be recognized. This serves to as needs be caution the drivers and prohibit specific activities like overspeeding. In this manner guaranteeing and keeping a beware of the traffic signs and likewise following them. Traffic signs without a doubt give us a huge number of data and guide us as needs be so we can move securely. Traffic Sign Classification is exceptionally valuable in Automatic Driver Assistance Systems.The proposed approach uses a custom convolutional neural network for training that supports in the identification and categorization of traffic sign images. To increase the accurateness of a given dataset, a set of classes are created and educated. We used the German Traffic Sign Benchmarks Dataset that includes 39,209 photos of traffic signs in about 43 categories. About 98.52 percent of the execution was accurate.

Literature Review

The Indian Traffic Sign Board Identification and Driver Alert System uses machine learning to identify traffic signs and alert drivers. There are numerous of important applications in domains such as sophisticated driver assistance systems, road surveys, and autonomous vehicles. Shubham Yadav, Anuj Patwa, Saiprasad Rane, and Chhaya Narvekar are the authors. To separate pertinent information from the real-time streaming video, our system uses an image processing technique. The five components of the suggested method are data gathering, data processing, data categorization, training, and testing.

To improve image quality, eliminate useless pixels, and identify edges, the system employs a number of image processing algorithms. The features in the image are located using feature extractors. Support Vector Machine (SVM), a machine learning technique, is used to categories the images based on their attributes. The speech signal will be generated to warn the vehicle if the video's captured attributes of the sign match those of taught traffic signs. Different traffic sign boards may be found in India, and they are divided into three groups: regulatory signs, cautionary signs, and informational signs. These Indian symbols come in eight distinct colors and four different shapes. The suggested system has been taught to recognize ten different sign kinds. More in everygroupingadditional than a thousand sample images are worn to train the network.

CNN's Real-Time Traffic Sign Recognition Design. AUTHORS: P. Yakimov and Shustanov

The system proposed by the authors for recognising and detecting road signs uses an all together of convolutional neural networks (CNNs) for visual processing. The CNN has a very high recognition rate, which raises its appeal for a variety of computer-based vision tasks. TensorFlow is the technique employed for CNN execution. Using German data sets, the authors of this paper were able to produce circular signs with accuracy greater than 99 percent.

A Colour Segmentation, Shape Matching, and SVM-Based Automatic Traffic Sign Detection and Recognition System. Safat B. Wali, Mahammad A. Hannan, Aini Hussain, and Salina A. Samad are the authors. Here, the authors explain the process they followed to put a brand-new approach to sign identification into practise. They fitted this cutting-edge ARK-2121 technology, a tiny computer, on the vehicle. SVM and HOG were the two main algorithms used in the sign's recognition phase. Their detection accuracy was 91%, and their average classification accuracy was at 98%.

Traffic Symbol Recognition Using Shape Analysis And K-Means. Thakur Pankaj D. and Manoj E. Patil are the authors. The colour image segmentation and integrated form analysis are the foundation of the suggested traffic symbol recognition system. The system that is being shown is reliable and capable of identifying traffic symbols in any colour and any existing shape (such as circular, rectangular, triangular, pentagonal, and octagonal), as well as being invariant to transformations like translation, scaling, and occlusion. The suggested system analyses various traffic symbol shapes using two separate algorithms, Peri2Area and boundary touching point, and then matches traffic symbol patterns using correlation coefficient.

The suggested method has three major phases: 1) segmentation, which involves grouping the pixels based on their colour properties to identify the ROIs; 2) Detecting traffic signs by utilising two innovative form classification criteria, namely the relationship between area and perimeter and the boundary touching point. 3) identifying the traffic symbol by using the correlation coefficient to compare the unknown symbols to the stored in the database known reference traffic symbols. The suggested framework offers a novel method for detecting a traffic symbol by fusing the geometric shape data with visual attributes.

EXISTING SYSTEM

In this day and age identification of traffic signs has turned into a significant part of our lives. Taking a gander at the rising traffic to guarantee the security of all and for programmed driving from now on traffic sign order is most extremely essential. The impressive examination has been finished around the acknowledgment of traffic and street signs. The precedingtechniques worn for designing the traffic sign recognition model are 1) K-means clustering 2) Lidar and vision based 3) Video streaming also in 1987 the primary exploration on the point "Traffic Sign Recognition" was finished by Akatsuka and Imai where they attempted to construct a major framework that could perceive traffic signs and caution the drivers and guarantee his/her security. Yet this was utilized to give the programmed acknowledgment to just some particular traffic signs. Traffic sign acknowledgment at first showed up as just speed limit acknowledgment in 2008. These images could identify the round speed limit signs. Then again later frameworks were planned that performed discovery on surpassing signs. This innovation was accessible in the Volkswagen Phaeton and in the 2012 in Volvo S80 V70 and some more.

PROPOSED SYSTEM

The Road Sign Board Recognition and Voice Alert System utilizing Custom Convolutional Neural Network is encouraged in the suggested framework. The driver would benefit greatly from our framework that is prepared to recognize, perceive, and interpret the street traffic signs. A programmed street signs recognition framework's objective is to identify and describe at least one street sign from among live variety images. Using the voice of the prestigious sign board, we inform the driver about the sign. Dataset The German Traffic Sign Benchmarks (GTSRB) Dataset, that consists of 39,209 images of traffic signs in roughly 43 categories, is used in the suggested framework.

SYSTEM ARCHITECTURE

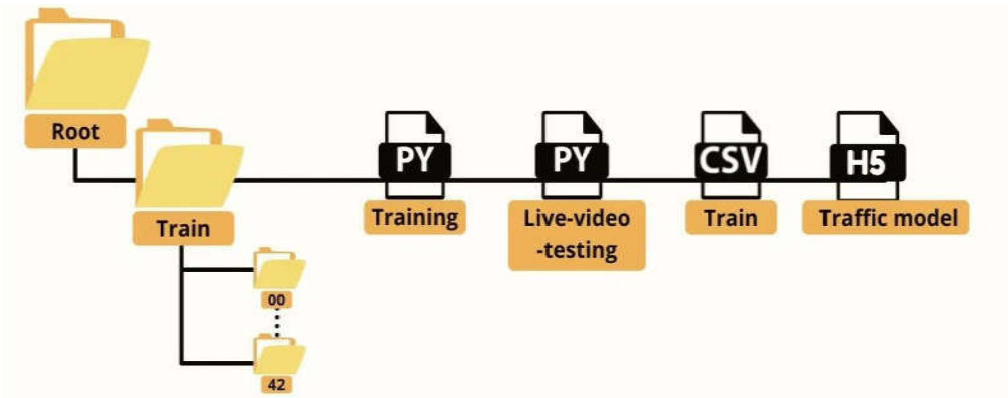

Fig.1. Software system architecture

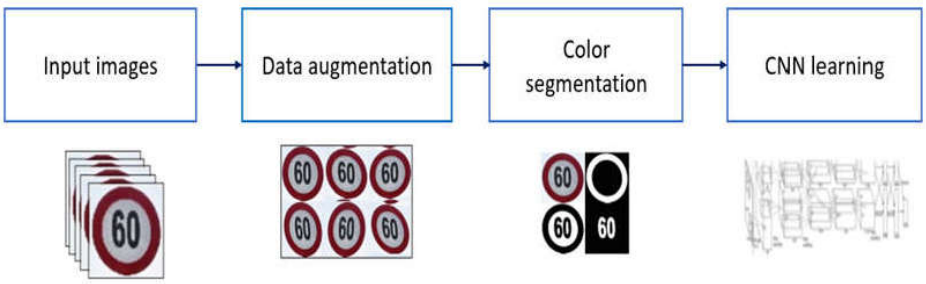

Fig.2. System Architecture of Training

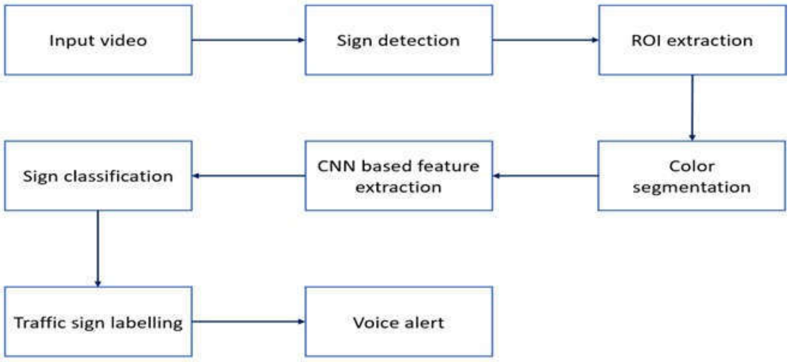

Fig.3. Overview of Traffic Sign Recognition Architecture

People generally lack the ability to read traffic signs in this time of a fast-paced life, which leads them to breach the law. In an effort to cut down on accidents, certain research has been removed from this area. To categories the traffic signs and alert the motorist, analysts have used a variety of classification algorithms and many CNN models. Our technology aims to improve the way of acknowledgment while also providing other advantages, such giving the driver a head start.

IMPLEMENTATION

Open CV:

Although it is published under a BSD licence, OpenCV (Open-Source Computer Vision Library) is accessible for both educational and commercial use. It has C++, Python, and Java interfaces and supports Windows, Linux, Mac OS, iOS, and Android. Having applications that operate in real time and optimum processing speed in mind, OpenCV was developed. The library can benefit from multi-core processing because it was written in optimised C/C++. It can benefit from the hardware rushing of the essential heterogeneous compute platform when OpenCL is enabled. It is a software library for computer vision and machine learning that is open source.

Keras:

On top of the Tensorflow machine learning platform, Keras is a Python deep learning API."Keras" is:

Simple -- but not too simple. Keras lessens the cognitive strain on developers so you may deliberate on the criticalfeature of the problem.

Flexible -- Due to the continuous exposure of complication philosophy espoused by Keras, simple processes ought to be quick and simple, but arbitrary complicated processes must be possible through an obvious route that builds against what you've previously learnt.

Strong -- Keras is employed by companies and organizations including NASA, YouTube, and Waymo because it provides speed and capacity that are unparalleled in the marketplace.

Sklearn: Scikit-learn is mostly written in Python and significantly makes use of the NumPy module for computations involving arrays and linear algebra. To further the effectiveness of this library, some basic algorithms are also written in Cython. Utilizing wrappers created in Cython for LIBSVM and LIBLINEAR, support vector machines, logistic regression, and linear SVMs are done. In certain conditions, expanding these functions with Python might not be practical.

Tensorflow:

TensorFlow, an open-source, free framework, is utilized for dataflow and distinguishable programming across a range of tasks. It is a symbolic math framework that is also used by machine learning programmers that use neural networks. It is utilized by Google for purposes of research and manufacturing. To work in machine learning, it is a requirement in the industry to have TensorFlow experience. The Google Brain team created

TensorFlow for usage within Google. On November 9, 2015, it was made available under the Apache 2.0 open-source licence.

TensorFlow may operate on an assortment of CPUs and GPUs thanks to available CUDA and SYCL enhancements enabling general-purpose computation on graphics processors. TensorFlow is supported by Linux 64-bit, macOS, Windows, as well as mobile operating systems including Android and iOS. Computing might be easily distributed across a variety of platforms (CPUs, GPUs, and TPUs) including PCs, server clusters, mobile devices, and edge devices thanks to its modular design. TensorFlow calculations are represented by autonomous dataflow graphs.

Pandas:

Pandas serve as an open-source Python toolkit that provides outstanding performance data analysis and manipulation tools utilizing its powerful data structures. Python was primarily employed for data munging and preprocessing. On analysis of data, it had little of an effect. Pandas discovered the answer. No matter where the data came from, we may use Pandas to carry out the five typical steps of data processing and analysis: prepare, modify, model, and evaluate. Many academic and professional fields, including finance, economics, statistics, analytics, etc., use Python with Pandas.

Numpy:

NumPy is a general-purpose library for managing arrays. It provides an extremely quick multidimensional array object and also the ability to interact with such arrays. This Python package is fundamental to scientific computing.

Matplotlib:

Publication-quality graphics are produced in a variety of tangible formats and cross-platform interactive environments using the Python 2D plotting package Matplotlib. Matplotlib can be used with several graphical programming toolkits, the Python and IPython shells, the Jupyter notebook, web application servers, and Python scripts. Matplotlib tries to render simple things straightforward while making challenging things doable. You can make graphs, histograms, power spectra, bar charts, error charts, scatter plots, and many more with merely a few lines of code. For instances, view the sample plots and thumbnail galleries. The pyplot package provides a MATLAB-like interface for simple plotting, especially when utilized in the context of IPython. Power users are given total control over line styles, font settings, axis characteristics, etc. through an object-oriented approach or other technique.

Seaborn:

Plotting statistical representations is made incredibly easy with Python's Seaborn visualizing module. It provides excellent preset color schemes and styles to improve the visual appeal of statistics charts. It has a Matplotlib software foundation and is closely related to Pandas data structures. Through Seaborn, data exploration and understanding shall be centered on visualization. It provides dataset-oriented APIs that let us transition among various visualizations for identical variables for greater comprehension of the dataset.

ALGORITHMS

A CNN is a Deep Learning algorithm that can discriminate among different things in an input image by assigning different attributes and entities in the image significance (learnable weights and biases). In comparison to other categorization methods, a ConvNet needs significantly less pre-processing. Contrary to earlier approaches, where filters must be hand-engineered, ConvNets are capable of learning these filters and their attributes. The organisation of the Visual Cortex and the connectivity network of neurons in the human brain both have a bearing on the design of a ConvNet. Individual neurons only respond to stimuli in this restricted region of the visual field, defined as the Receptive Field. These fields combine in an assortment to fill the whole visual field.

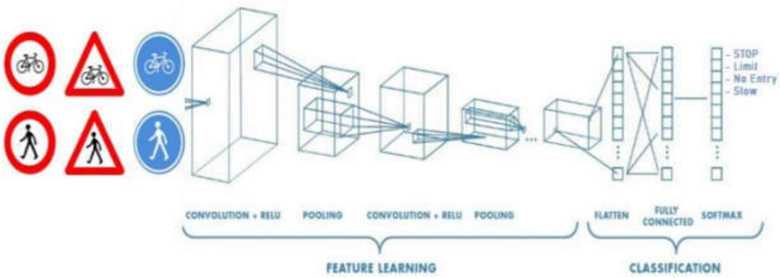

Fig.4. Architecture of CNN

Custom CNNs:

Custom CNNs allow you to design and customize the structural design of a Convolutional Neural Network according to the specific requirements of your task. Here's some information to help you understand and create a custom CNN:

a. **Convolutional Layers:** The convolutional layers are the core building blocks of a CNN. They relate filters to input data, typically images, to remove features. Each filter performs a convolution operation by sliding over the input and computing dot products with the local receptive field. The amount of filters determines the depth of the output feature maps.
b. **DropOut Layer:**Dropout is a method for avoiding excessive fitting in models. Throughout training, the Dropping out layer arbitrarily sets input units to 0 at a frequency of rates at every stage, which helps prevent overfitting and thus making the model generalize better.
c. **Pooling Layers:** Pooling layers are often worn after convolutional layers to diminish the spatial dimensions of the feature maps while retaining important information. The most common type is max pooling, which takes the maximum value from a region of the feature map and discards the rest.
d. **Activation Functions:** Activation functions bring in non-linearities into the CNN, permiting it to learn complex patterns and make non-linear decisions. Popular commencement functions comprise ReLU (Rectified Linear Unit), sigmoid, and tanh. ReLU is commonly used due to its efficiency and ability to alleviate the vanishing gradient problem.
e. **Fully Connected Layers:**subsequent to the convolutional and pooling layers, the feature maps are flattened into a 1-dimensional vector and passed through fully connected layers. These layers connect each neuron as of the preceding layer to the present layer, allowing the network to learn high-level representations and perform the final classification.
f. **Loss Function:** The alternative of a loss function depends on the task at hand. For classification problems, common choices include softmax cross-entropy loss for multi-class classification or binary cross-entropy loss for binary categorization. The loss function quantifies the difference between the predicted output and the true labels, which the network aims to minimize during training.
g. **Optimization and Training:** Training a CNN involves an optimization process that updates the network's weights to minimize the loss function. Popular optimization algorithms comprise Stochastic Gradient Descent (SGD), Adam, and RMSprop. The training process involves feeding batches of labeled data to the network, computing the loss, and backpropagating the gradients to update the weights.
h. **Hyperparameter Tuning:** When creating a custom CNN, you have control over several hyperparameters that can affect the network's performance. These include the number and size of filters, the size of pooling operations, the depth and width of the network, learning rate, batch size, and more. Tuning these hyperparameters can significantly impact the CNN's ability to learn and generalize from the data.

Custom CNNs offer flexibility and adaptability for various computer vision tasks. You can experiment with different architectures, layer configurations, and hyperparameters to achieve optimal results based on the specifics of your dataset and problem domain.

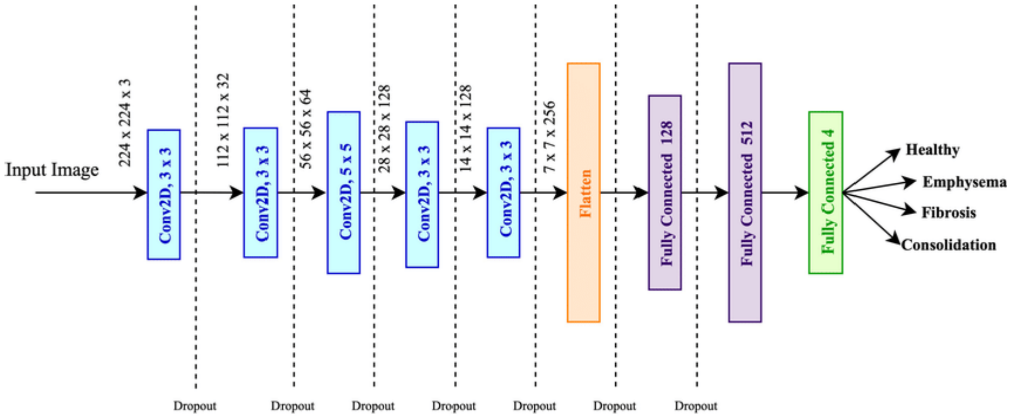

Fig.5. Custom CNN Architecture

STEPS IN CUSTOM CNN

1. **Dataset:** The "German Traffic Sign Recognition Benchmark" dataset was utilised. Only the "Train Images" folder and the train.csv file were utilised for training. Dataset has 39,209 total photos in 43 classifications.
2. **Import Libraries:** Import all required libraries first, including opencv (to read photos from the local disc), keras, tensorflow (to build neural networks), sklearn (to divide data into training and testing and for confusion matrix), pandas, seaborn, and random (among others).
3. **Read Images:** Read the photographs from the local drive, resize them to (32x32), and then add them to the list along with the label that goes with them. The size of each image should be same.

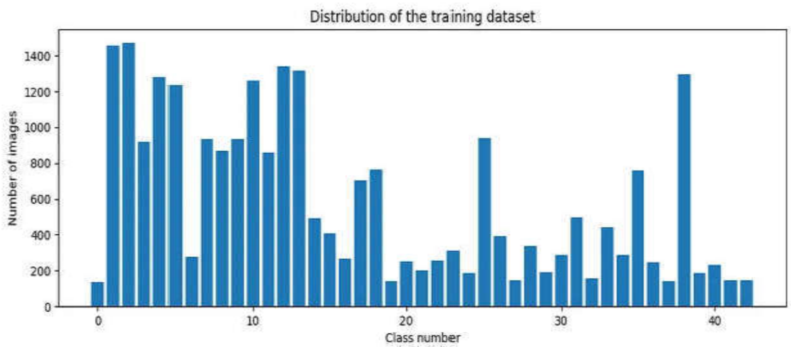

Fig.6. Dataset Distribution

Fig.7. Various Traffic Sign's

To prepare the image data for the model's input, first turn the list into an array and then resize it to (32x32x3). The image is (32x32) pixels in size, and its RGB (Red, Blue, and Green) channel is 3.

Splitting dataset: creating train, test, and validation datasets. 60% of the data used to train, 20% to test, and 20% to validate.

Preprocessing: Apply some image processing, such as grayscale conversion, histogram equalisation, and picture normalisation, to the training, validation, and test images.

Data Augmentation: Data augmentation has been used to provide more generic data. We have established many criteria for the augmentation, such as a 10% range for width and height shifts, a 0.2 range for zooming, and a 10 degree range for rotation. 20 photos are augmented simultaneously if batch_size=20.

Create Model: We utilised a sequential model from the Keras library to build the model. Convolutional layers must then be added to the network. We utilised 60 filters in the first two Conv2D layers, and the kernel size was 5,5. The image features are collected by this kernel as it goes around the image. Pool size (2,2) was utilised in MaxPooling2D.

training

Model

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 60)	1560
conv2d_1 (Conv2D)	(None, 24, 24, 60)	90060
max_pooling2d (MaxPooling2D)	(None, 12, 12, 60)	0
conv2d_2 (Conv2D)	(None, 10, 10, 30)	16230
conv2d_3 (Conv2D)	(None, 8, 8, 30)	8130
max_pooling2d_1 (MaxPooling2	(None, 4, 4, 30)	0
dropout (Dropout)	(None, 4, 4, 30)	0
flatten (Flatten)	(None, 480)	0
dense (Dense)	(None, 500)	240500
dropout_1 (Dropout)	(None, 500)	0
dense_1 (Dense)	(None, 43)	21543

Total params: 378,023
Trainable params: 378,023
Non-trainable params: 0

Fig.8. Model Summary

It constitutes one of two components of the software testing strategy known as "box testing". Blackbox evaluation, which is its opposite, entails testing using an outside or end-user viewpoint. Whitebox evaluation, on the other hand, focuses on internal evaluation and is focused on an application's internal functioning. Due to the idea of a see-through box, the name "whitebox" was adopted. The phrase "clear box" or "whitebox" refers to the ability to see via the software's outer layer (or "box"). Similarly, the "black box" in "black box testing" stands for the inability to examine the insides of the software in order to test just the user exper ence as a whole.

S.NO	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Success/ Failure**
1.	Check whether a sign board is detected	Test with GTSRB datasets	Figure 7.1	Successful in displaying the result	Success
2.	Check whether a sign board is recognized or not.	Test without GTSRB datasets	Figure 7.4	Successful in displaying the result	Success

Table.1. Testing the dataset with and without sign boards

S.NO	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Success/ Failure**
1.	Check whether a sign board is detected or not	Capture image	Figure 7.1	Successful in displaying the result	Success
2.	Check whether a sign board is recognized or not.	Live Stream the video in public places	Figure 7.4	Successful in displaying the result	Success

Table.2. Testing with live streaming and image capturing.

RESULTS

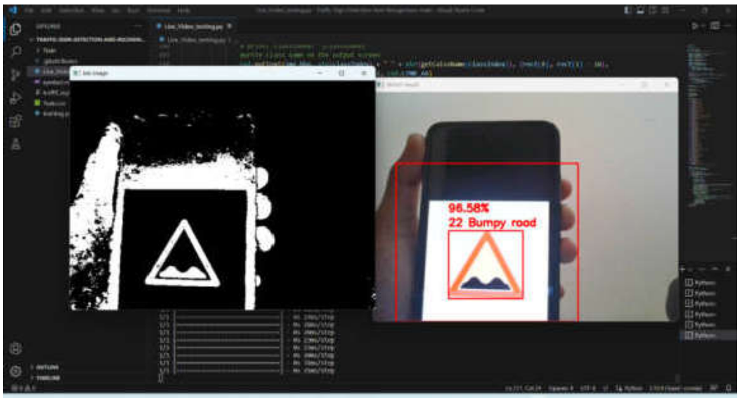

Fig.9. Detection of Bumpy Road

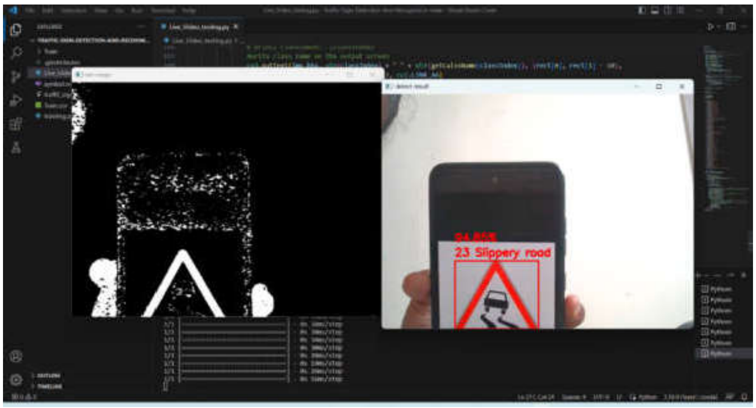

Fig.10. Detection of Slippery Road

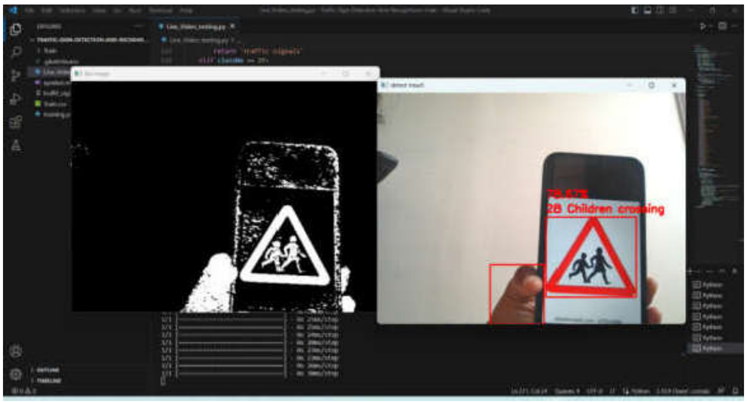

Fig.11. Detection of Children Crossing

Fig.12. Detection of Speed Limit in Real Time

CONCLUSION

In conclusion, our project focused on developing a Traffic Sign Board identification and Voice Alert System using a Custom Convolutional Neural Network (CNN) algorithm. The system leverages the German Traffic Sign Recognition Benchmark (GTSRB) dataset to train the model, enabling it to detect sign boards in real-time using a webcam. By providing voice notifications to the user, the system enhances driver awareness and promotes road safety.Through the utilization of a custom CNN algorithm, we achieved accurate and efficient recognition of traffic sign boards. The GTSRB dataset served as a valuable resource for training the model, as it contains a diverse range of traffic sign images, encompassing various angles, lighting conditions, and weather scenarios.

This enabled the model to generalize well and accurately classify different types of signs encountered in real-world scenarios.
The system's implementation involved a real-time video processing pipeline that continuously captured frames from the webcam. Each frame was processed using the custom CNN model, which performed object detection and classification to identify the presence of traffic signs. Upon recognition, the system generated voice alerts to inform the user about the meaning and instructions associated with the detected sign.To assess the system's presentation, we estimate its accurateness, exactness, recall, and F1 score. The evaluation process involved testing the system under different lighting conditions, weather scenarios, and sign orientations to ensure its robustness and reliability. The consequencesverified the efficiency of the custom CNN model in accurately recognizing and classifying traffic sign boards in real-time.
The developed Traffic Sign Board Recognition and Voice Alert System contribute significantly to road safety. By providing voice notifications, the system enables drivers to receive timely and relevant information about traffic signs, allowing them to make informed decisions on the road. This helps minimize the risk of accidents and violations by ensuring compliance with traffic regulations.

FUTURE SCOPE

The future scope for Traffic Sign Board Recognition and Voice Alert System using Custom CNN is promising and includes several potential advancements:
Multilingual Support: Expanding the system's voice alert capabilities to support multiple languages can improve its usability and applicability in different regions. This would involve developing multilingual text-to-speech synthesis models and integrating language translation algorithms to convey the meaning of traffic signs accurately.
Integration with Autonomous Vehicles: Traffic sign recognition systems can be integrated into autonomous vehicles to enhance their perception capabilities. By accurately detecting and interpreting traffic signs, the system can contribute to safer decision-making and improve overall autonomous driving performance.
Collaborative Data Collection: Encouraging the involvement of the community in data collection efforts can help expand and enrich the existing traffic sign databases. Crowdsourcing initiatives or mobile applications can allow users to contribute images of traffic signs, thereby enhancing the training and validation datasets.
Integration with Traffic Management Systems: Integrating the Traffic Sign Board Recognition and Voice Alert System with existing traffic management systems can facilitate better traffic control and management. By providing real-time information about detected signs, the system can contribute to intelligent traffic signal control and optimize traffic flow.

REFERENCES

[1]. V. Ciuntu and H. Ferdowsi, "Real-Time Traffic Sign Detection and Classification Using Machine Learning and Optical Character Recognition," 2020 IEEE International Conference on Electro Information Technology (EIT), 2020, pp. 480- 486, doi: 10.1109/EIT48999.2020.9208309..
[2]. S. Harini, V. Abhiram, R. Hegde, B. D. D. Samarth, S. A. Shreyas and K. H. Gowranga, "A smart driver alert system for vehicle traffic using image detection and recognition technique," 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India,2017,pp.15401543,doi:10.1109/RTEICT.2017.8256856
[3]. Fleyeh, H. and Dougherty, M., "Road and Traffic Sign Detection and Recognition" 10th EWGT Meeting and 16th Mini-EURO Conference, Poznan, Poland, 13-16 September,2005.
[4]. A. Welzel, A. Auerswald and G. Wanielik, " Accurate camerabased traffic sign localization," 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 2014, pp. 445-450, doi: 10.1109/ITSC.2014.6957730.

[5]. Saha S., Islam M.S., Khaled M.A.B., Tairin S. (2019) An Efficient Traffic Sign Recognition Approach Using a Novel Deep Neural Network Selection Architecture. In:Abraham A., Dutta P., Mandal J., Bhattacharya A., Dutta S. (eds) Emerging Technologies in Data Mining and Information Security. Advances in Intelligent Systems and Computing, vol 814. Springer, Singapore. https://doi.org/10.1007/978-981-13-1501- 5_74

[6]. R. Qian, Y. Yue, F. Coenen and B. Zhang, "Traffic sign recognitionwith convolutional neural network based on max pooling positions,"2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Changsha, China, 2016, pp. 578- 582, doi: 10.1109/FSKD.2016.7603237.

[7]. H. S. Lee and K. Kim, "Simultaneous Traffic Sign Detection and Boundary Estimation Using Convolutional Neural Network," in IEEETransactions on Intelligent Transportation Systems, vol. 19, no. 5, pp.1652-1663, May 2018, doi: 10.1109/TITS.2018.2801560.

[8]. Dr. G. Kalpana,et al, Machine Learning Model with Hybrid Ensemble Feature on Estimating Different Word Count Prediction Model, Volume 12, Issue 04, Apr 2022,IJARST.ISSN:2457-0362.DOI: 10.48047/IJARST/V12/I04/02

[9]. Chuanwei Zhang et al., Study on Traffic Sign Recognition by Optimized Lenet-5 Algorithm, International Journal of Pattern Recognition and ArtificialIntelligence, doi:0.1142/S0218001420550034.

[10]. S. Huang, H. Lin and C. Chang, "An in-car camera system for traffic sign detection and recognition," 2017 Joint 17th World Congress ofInternational Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSASCIS),Otsu, Japan, 2017, pp. 1-6, doi: 10.1109/IFSASCIS.2017.8023239.