[1]Dr. Rahul Patil

[2]Dr. Prashant Ahire

[3]Dr. Kalyan Bamane

[4]Dr. Abhijit Patankar

[5]Dr. Pramod D. Patil

[6]Saomya Badoniya

[7]Resham Desai

[8]Gautam Bhandari

[9]Bikramjeet Singh Dhami

Real-Time Traffic Sign Detection and Recognition System using Computer Vision and Machine Learning

Journal of Electrical Systems

Abstract: - The use of computer vision technology for our system has taken a huge leap and has revolutionized the way in which road safety is managed and maintained. This part of the paper represents the key components and functionalities of such a system which helps in enhancing the road safety and also in traffic management. The proposed system is meant to contain computer vision algorithms and some machine learning techniques. These will help in detecting and recognizing the traffic signs in real time video streams. With the help of deep learning models like convolutional neural networks, the proposed system can precisely identify various traffic signs present on roads. The major components of the proposed system include a camera interface which will capture real time video frames from the road, image processing modules for color space conversions, binary thresholding, contour detection and a classification module for the purpose of sign recognition. Deep learning frameworks like TensorFlow and OpenCV libraries are used for image processing and model training In the proposed system. The real time nature of this system allows the spontaneous detection and recognition of traffic signs enabling us to change the road conditions. Hence, the system contributes to prevent accidents and provide timely warnings to drivers. In addition, this system offers potential applications in traffic management and for autonomous driving system. With the help of accurate information, the system can contribute towards more efficient traffic flow and reduce congestion on roads. Overall, the system has very good impacts in road safety and efficiency. It shows advancement in road safety and traffic management.

Keywords: Computer vision, TensorFlow, Keras, OpenCV, real time video, color space conversion, binary threshold, contour, streamlit, object detection, classification.

I. INTRODUCTION

A real-time traffic sign detection and recognition system using computer vision is a challenging research problem in the field of road safety. For the purpose of enhancing road safety and optimizing traffic flow, there is a need for accurate and real-time recognition of traffic signs. Fields like deep learning and computer vision have recently experienced huge advancements by enabling the development of sophisticated traffic sign detection and recognition systems. This research paper gives insights about a comprehensive review on state - of-the-art research in real-time system

[1]Asst. Prof., Pimpri Chinchwad College of Engieering, patil.rahul3068@gmail.com

[2] Asst. Prof., Pimpri Chinchwad College of Engieering, prasshantahire@gmail.com

[3] Asso. Prof., D.Y Patil College of Engineering, kdbamane@dypcoeakurdi.ac.in

[4]Asso. Prof., D.Y Patil College of Engineering, ajpatankar@dypcoeakurdi.ac.in

[5]Prof., Dr. D.Y Patil Institute of Technology, pdpatiljune@gmail.com

[6] Saomya Badoniya:Pimpri Chinchwad College of Engieering

[7] Resham Desai:Pimpri Chinchwad College of Engieering

[8] Gautam Bhandari:Pimpri Chinchwad College of Engieering

[9]Bikramjeet Singh Dhami:Pimpri Chinchwad College of Engieering

The author [1] has proposed a neural network-based approach to high-definition images for the identification of road traffic signs. It demonstrates superior performance in real-time applications where it utilizes techniques like parallelization and region focusing. An introduction to evaluation methods [2] for deep learning-based embedded systems used in traffic sign detection is shown here. The role of achieving real-time processing and hardware acceleration is emphasized here. The use of YOLO V4 for generating training data with the help of GaMs for leveraging synthetic training data and for traffic sign recognition is depicted here [3]. The approach represents the efficacious nature of synthetic data for the purpose of improving the performance of the generated model. With the intention of focusing on enhancing the process of object detection performance, especially for smaller signs, real-time detection was developed based on YOLO V3. [4] The method was for small traffic signs. Later, a system was proposed for light-weight traffic sign recognition where a light-weight traffic sign recognition algorithm which is based on YOLO V4-Tiny was used. It helped in achieving a proper balance between the performance and complexity of the model[5]. [8] For showcasing the potential of ensemble methods for improving accuracy, the author explored traffic sign recognition using CNN ensemble. With respect to implementation, various libraries and frameworks, like Keras and TensorFlow, are used. They helped in training and building deep learning models. The commonly used frame work for computer vision tasks and for image processing is open CV. Additionally, to improve feedback and user interaction features such as beep sounds when a sign is detected or providing real time alerts to drivers. By using streamlit, we can easily create a user friendly interface which will allow users to interact with the application effortlessly.

II. Related Work

Several studies have been published on traffic sign detection and classification with static datasets; however, implementing it in real time is a complex research challenge. Various strategies for processing traffic signs were developed and designed during the previous decade for solving the problem of road accidents and for traffic management. Hereby we evaluated the previous methods and created a system where we could process real time data and give outputs accordingly. We have managed to create a system where the driver will be notified through a beep sound and a message whenever a road traffic sign is detected.

A. State-of-the-arts

The authors of [1] have explored various old or previously stated methods for the detection and recognition of traffic signs with the help of neural networks. They have focused in approaches that deal with high quality or high-definition images. They have also used parallelization techniques for the same. The authors have also mentioned the limitations of the paper as compared to previous old methods. In [2] research or investigation about the already prevailing methods for traffic sign detection based on deep learning systems are discussed. It emphasizes on the importance of embedded systems. Various techniques and evaluation methodologies used in previous research papers have been discussed that help assess the performance of such systems.

The authors of [3] have reviewed the use of YOLOv4 and the use of synthetic training data that has been developed or generated by GAN in computer vision for advanced traffic sign recognition. The paper displays the advantages and the challenges that are aligned with this approach. In [4] the authors have compared the performance of YOLOv3 based systems with other object detection approaches. They have considered parameters like speed, accuracy, especially for small object detection. The authors have also examined the previous real time detection methods that can be used for small traffic signs which basically focus on the proper utilization of YOLOv3.

The authors of [5] have explored the already available ways for developing a lightweight traffic sign recognition algorithm. These algorithms are especially based on YOLOv4-Tiny. Detailed discussion on prior research where they have discussed about optimizing various deep learning models that are used for resource-constrained environments and are used for significantly highlighting the improvements that are achieved by this algorithm as compared to the previous ones. In [6] the authors have researched about the previously available work on the same with the help of convolutional neural networks (CNNs). They have done comparison between various CNN architectures and the already stated methodologies that are used for this task. Also emphasis on approaches that prioritize computational efficiency and accuracy are given.

The authors of [7] have discussed prior research on applications of AI and deep learning In the field of traffic sign recognition tasks and mentions about the architecture performances. It also mentions about the review of existing deep learning algorithm, focusing on convolutional neural network (CNNs).In [8] the authors have discussed about the benefits for attention mechanism. It helps in enhancing performance and interoperability by use of CNNs for task compared to traditional architecture. The paper also explains recent inventions in attention-based deep convolutional neural networks(CNNs) for real time traffic sign recognition in smart vehicles. The author [9] investigates the earlier studies on traffic sign classification with the help of CNN and computer vision techniques. It explains various techniques for feature extraction and for classification of algorithms mentioned in pervious searches.

In [10] the challenges and specifications for implementing traffic sign detection system have been discussed in automotive environments which also includes issues related to accuracy and performance.The author [11] emphasizes the benefits of combing multiple models for improvising the performance and accuracy of the established system.

B. Research Gap Analysis

In the above section, we have reviewed the various recently proposed real time traffic sign detection and recognition methods under different categories. We have seen the use of various modules like TensorFlow, Keras as well as CNNs etc.

In [1],a system has been proposed which uses region focusing and parallelization. The paper lacks a comprehensive discussion on the limitations and challenges faced during the implementation of the proposed system. This includes issues related to dataset annotation, model training, and real-time deployment. The gap is in understanding how these techniques can be used for future applications in various scenarios.

A potential gap or lacking found in [2] could be that a deeper analysis of the trade offs between accuracy, speed etc. in such embedded systems. This paper discusses deep learning based system for traffic sign classification and recognition.The paper [3] discusses about using synthetic training data which Is generated with the help of GANs .So, this could lead to a gap in understanding how the models are trained on synthetic data with respect to real world scenarios.

In paper [4] there might be a gap while checking the robustness of methods based on YOLOv3 with varying lighting and weather conditions. The paper [5] explains about light weight traffic sign recognition algorithm which is based on YOLOv4-Tiny.One potential gap in this system could be the comparison of this algorithm with the existing methods in terms of accuracy and efficiency. In the paper [6] the authors discuss about use of CNNs for real time traffic sign detection and recognition. One potential gap in this system is that there can be exploration of newer CNN architecture for improving the accuracy, efficiency and performance of the system.

The paper [7] represents an automatic traffic sign recognition algorithm which is based on deep learning. The gap could be that we might not understand the scalability and adaptability of the algorithms that are used.A potential gap in [8] could be that we cannot do analysis of the cost and real world deployment challenges of CNN methods used. In paper [9] we cannot investigate the effectiveness of ensemble learning for improving the robustness of the traffic sign recognition systems.

The paper [10] talks about traffic sign classification using CNNs and computer vision. The drawback of this paper is that we cannot explore approaches for integrating CNNs with computer vision.In paper [11] we get information about a real time traffic sign detection and recognition system for in-car driver assistance systems. A potential gap in this paper could be the analysis of the systems performance in real world driving scenarios.In paper [12] the gap in the literature presents an opportunity to enhance the functionality of existing systems by integrating notification capabilities, which can contribute to improved road safety and driver awareness.These gap analysis can help identify areas where further research or improvements are needed in the field of real-time traffic sign recognition using computer vision.

C. Contributions

Our contribution to some of the research gaps which we have studied from our related works is that –

1) We have implemented a traffic sign recognition system which functions in real time with a basic UI designed by streamlit. Many of the research papers who implemented the system of traffic sign detection lacked to deploy it in real time which is a major contribution from our system. Practical applications require real-time functioning, especially when it comes to traffic safety systems where timely information is critical for making decisions. Users can engage with the system more easily because of the increased accessibility and usability brought about by the usage of Streamlit in the user interface (UI) design.

2) We have also integrated the notification alert system which is a gap in [12]. When our system recognizes the traffic sign it sends an alert to the driver's phone number as an SMS. This can be implemented by using Twilio API which is a cloud communication platform for sending and receiving SMS, voice calls etc.

III. SYSTEM ARCHITECTURE

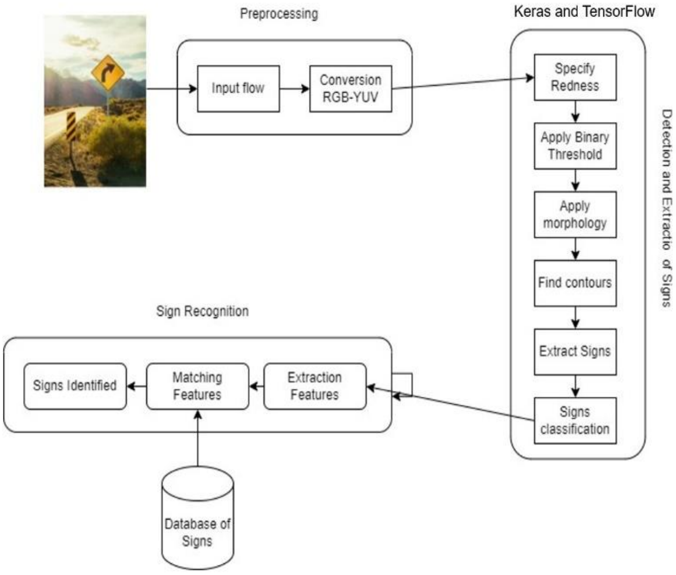


Figure 1 System Architecture

1) Input: It uses live video from a camera, looking at the road and signs.

2) Preprocessing: First, it adjusts the colors to focus on red, which is common for signs. This makes it easier to spot signs.

3) Feature Extraction: It then looks for the shapes of signs in the video. This helps find where the signs are.

4) Sign Classification: Next, it tries to figure out what each sign is by comparing them to signs it knows from a big list.

5) Feature Matching and Sign Identification: It matches the signs it finds with the ones it knows, making sure it's identifying them correctly.

6) Output: Finally, it shows or uses the information about the signs it found, like what type of sign it is.

7) Real-time Processing: All of this happens quickly, so it can keep up with the signs as they pass by.

8) 8. Scalability and Efficiency: The system is designed to work well even if there are lots of signs or the conditions are not perfect, while still being fast.

IV. Proposed System

We have proposed a system which detects and recognizes the traffic signs from real time video and generating an audio feedback and notification alert via SMS on driver's phone number. First, a large dataset of 4600 images of traffic signs in a variety of sizes, colors, and forms were gathered. In order to improve contrast and feature extraction, we preprocessed the dataset by using binary thresholding to isolate the traffic signs from the background noise, reducing the size of the images to a consistent resolution, and switching the color space from RGB to YUV. Morphological approaches such as erosion and dilation were also used to improve the segmentation results and ensure minimal distortion while isolating traffic signs.

The system was then trained for traffic sign recognition using a deep learning model we built with TensorFlow and Keras. The model framework comprised numerous convolutional layers, feature extraction through max-pooling layers, and classification by fully connected layers. In order to train the model with an optimal performance on a separate validation set, we tried different batch sizes and epoch numbers using an appropriate optimizer and a loss function. After 15 training epochs, the system achieved an incredible accuracy of 99.6% on the validation set.

In addition, we included real-time visualization and aural input into our system to enhance user engagement and situational awareness. Once a traffic sign has been correctly recognized in a real-time video feed, we dynamically generate a border box around it using computer vision libraries such as OpenCV. This border box assists in emphasizing the location and dimensions of the specified traffic sign within the video frame by providing the user with visual cues. We also added a simple sound producing device that, upon correctly identifying a traffic sign, will produce a short beep sound. This feature acts as an audible alarm to notify individuals when relevant signs are present nearby. An SMS notice on the identified traffic sign is delivered to the driver's phone number. User Interface has been developed using Streamlit, a library in python and integrated it with the our main module and model.

A. Problem statement and objectives

To develop a system which works in real time using computer vision and classification and involves creating a robust solution that can accurately identify and interpret various traffic signs from the input stream.

1) To collect road sign dataset and perform appropriate pre-processing.

2) To detect traffic signs from real-time video using Computer Vision.

3) To recognize detected traffic signs using Machine Learning model.

4) Facilitate regulatory compliance by ensuring the system recognizes and communicates the meanings of various traffic signs accurately.

B. Developing Tensorflow Keras Classifier Model

Neural network model created with the Keras API which is incorporated into the TensorFlow library is a TensorFlow Kerasmodel.High-level neural network API called Keras makes it easy and quick to experiment with deep learning models. It offers an intuitive user interface for creating, refining, and implementing many kinds of neural network topologies. On the other hand, Google's open-source TensorFlow machine learning library provides robust tools and resources for building and deploying machine learning models.When creating a neural network model, TensorFlow's computational capacity and Keras's ease of use can be combined with the help of the latter's Keras API. This combination makes it simple to build and train complex neural network topologies for a variety of machine learning uses, such as classification and regression.TheKeras model from TensorFlow leverages the Keras API to express the model architecture and training procedure, and it leverages the computational backend provided by TensorFlow for efficient execution on CPU or GPU hardware.

The sequential developed model consists of four layers which are –

- keras.layers.Conv2D: A 2D convolutional layer with 8 filters, each with a kernel size of 3X3, using ReLU activation function. The input shape of the layer is (28,28,1), indicating grayscale images of size 28x28 pixels.
- keras.layers.MaxPool2D: Max pooling layer with a pool size of (4,4).
- keras.layers.Flatten: Flattens the input, converting it into a one-dimensional tensor.
- keras.layers.Dense: A dense layer with 4 units and softmax activation function, representing the output layer for classification into 4 classes.

Adam optimizer is used for gradient descent optimization to minimize the loss and improve accuracy. The model is fitted with training images data which are 80% of the dataset and with epochs=15 and then tested on the 20% which gave final model evaluation as accuracy = 99.68%.

Algorithm: Model Developing

Inputs

D: Dataset of images
csv_file: CSV annoted files of each class of sign

Output

model: Developed TensorflowKeras classifier model

```
Importing necessary libraries like tensorflow, keras, sklearn etc.
Split training and testing data. Initial test size = 0.2
mu ← mean(D)
std ← std(D)
x_train_normalized← (x_train - mu)/std
x_test_normalized← (x_test - mu)/std
preprocessing()
class myCallBack(…):
    def on_epoch_end(…):
if(accuracy > 0.997):
        Stop model training
End if
model = keras.Sequential([…])
model.compile()
epochs = 15
callbacks = [myCallBack()]
Training the model using model.fit()
 For evaluation model.evaluate()
model.save('TSmodel')
model = keras.models.load_model('TSmodel')
```

A. System Operation Overview

1) Real time video is taken as an input for the system.

2) Multiple frames from the video are captured until the camera is stopped.

3) To specify redness in an image, a defined returnRedness() named function is used which converts the frame's RGB color space to YUV, splits and returns v. cvtColor() is a predefined function in OpenCV which is used for this purpose.

4) Applying binary thresholding to separate the traffic signs from the background noise. In binary thresholding, a threshold value is chosen, and each pixel in the image is compared against this threshold. If the pixel intensity is above the threshold, the pixel is assigned a value of 255 (white), indicating foreground or object of interest. If the pixel intensity is below the threshold, the pixel is assigned a value of 0 (black), representing background or irrelevant pixels.

5) Applying morphology takes an input image and performs a closing morphological operation on it using a kernel of size determined by kernelSize. The closing operation combines dilation followed by erosion, useful for closing small holes or gaps in foreground objects. The function returns the resulting image after the morphological operation.

6) The findContour function utilizes OpenCV's findContours function to extract contours from a binary image (img). The function returns the list of contours found in the image.

7) The findBiggestContour function takes a list of contours and calculates the area of each contour. It then returns the contour with the largest area, which represents the biggest object in the image.

8) Developed tensorflow and keras classifier model for the detection and recognition of the traffic signs.

9) Defined a boundarybox function to highlight the position and size of the identified traffic sign within the video frame.

10) Whenever, a traffic sign is recognized with the help of frames sent to model the system beeps with a frequency of 2500Hz and interval of 1 second as an audio feedback and the name of predicted sign is displayed in terminal.

11) Using twilio we create a notification alert which is sent via SMS on driver’s phone number.

V. Experimental Results

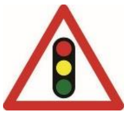
A. Test Case: Predicting the final recognized traffic sign according to the predicted percentage of each class

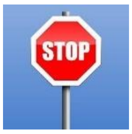
of signs.

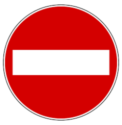
The final results of the test are provided in the below table.

Table 1. Prediction Percentage of each class of traffic sign

Input Traffic sign image	**Prediction % of each sign class**	**Final Result**

	Stop = 0.0004716%
	Do not enter = 6.217309e-15%
	Traffic jam = 99.99953%
	Yield = 1.8554727%
		Traffic jam is close sign

	Stop = 94.39929%
	Do not enter = 1.481e-05%
	Traffic jam = 3.0515%
	Yield = 2.5493%
		Stop Sign

	Stop = 3.57431%
	Do not enter = 99.996%
	Traffic jam = 6.0041e-13%
	Yield = 5.0043e-06%
		Do not Enter

	Stop = 0.014885%
	Do not enter = 7.1543e-08%
	Traffic jam = 98.7804%
	Yield = 1.2040309%
		Yield

B. Relationship between accuracy and loss with epochs

Finding the optimal number of epochs often involves experimentation and monitoring accuracy and loss (a measure of how well the model is learning) over epochs.

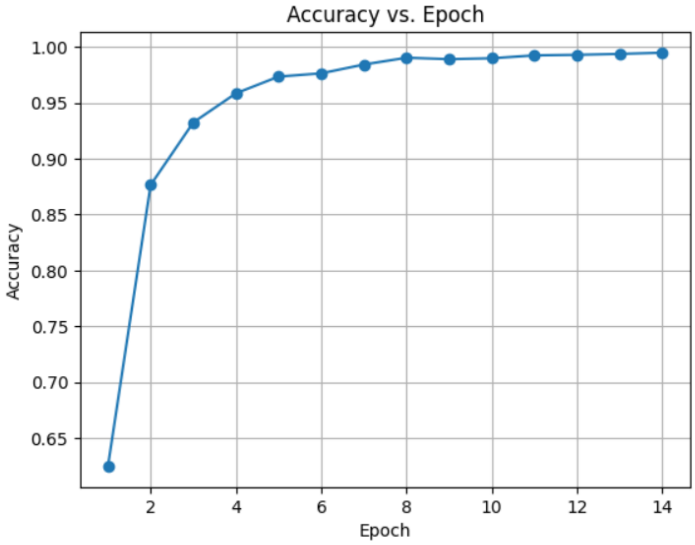


Figure 2 Accuracy vs epoch graph

From the above graph, it is concluded that accuracy is directly proportional to the number of epochsrequired to train a model. With each epoch, the model refines its understanding of the data and becomes better at recognizing patterns.

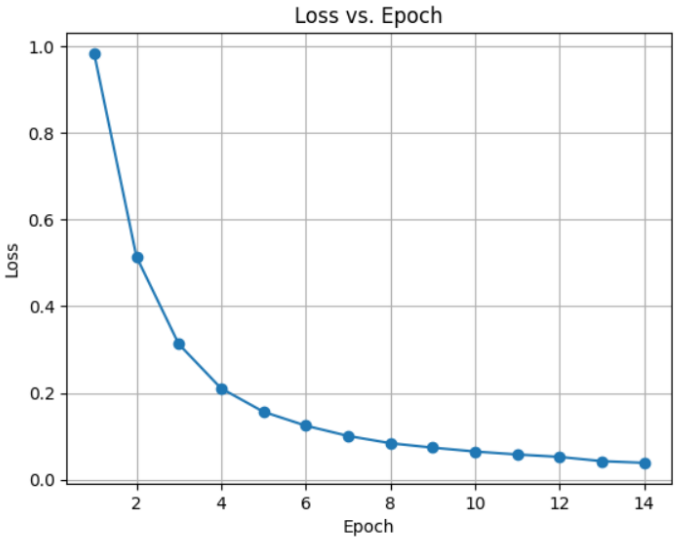


Figure 3 Loss vs epoch graph

From the above graph, it is concluded that loss is inversely proportional to the number of epochs. For a good fit of model, we have used adam optimizer which is a gradient descent optimization algorithm used to minimize the loss function.

A. Performance Metrics

Performance metrics are quantifiable measures which are used to assess how well a model performs on a specific task. For classification problems, confusion matrix is used to calculate recall, precision and F1 score which are the performance metrices.

$accuracy: (TP+TN)/(TP+FP+TN+FN)$ (1)

$precision: TP/(TP+FP)$ (2)

$recall: TP/(TP+FN)$ (3)

$f1: (2.precision.recall)/(precision+recall$ (4)

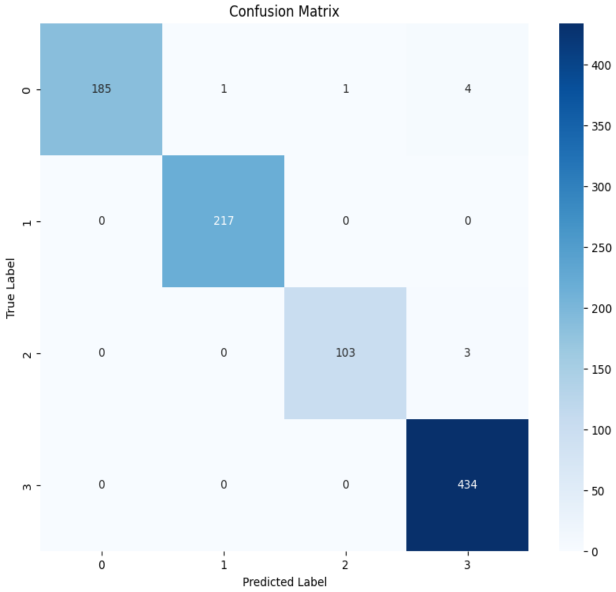


Figure 4 Confusion Matrix

From the above matrix, it is concluded that the confusion matrix reveals a strong performance by the model across all classes. The model appears to perform best on Class 0 with a high True Positive (TP) value of 434 and a low False Negative (FN) value of 0.

Table 2. Performance metrics for Keras classifier

Metric	Class 1	Class 2	Class 3	Class 4
Precision	0.9864	1.0000	0.9902	1.0000
Recall	1.0000	0.9954	0.9967	0.9841
F1 Score	0.9931	0.9977	0.9934	0.9920

- Class 1: The model achieved recall of 1.0000 which signifies that the model effectively captured all or almost all of the Class 1 data points. The high precision of 0.9864 indicates the vast majority of predictions for Class 1 were indeed correct.
- Class 2: The model demonstrated exceptional performance on Class 2 with a recall of 0.9954 and a precision of 1.0000. This suggests that the model rarely missed Class 2 instances and most of the predicted labels from Class 2 were accurate.
- Class 3: The model has a recall of 0.9967 and precision of 0.9902. These values suggests that the model effectively identified most of the Class 3 data points and the predictions were mostly accurate.
- Class 4: Comparing the other classes with class 4 model attained recall of 0.9841 and a precision of 1.0000 which indicates that the model was adept at classifying Class 4 instances and the predictions were highly accurate.

VI. CONCLUSION

Our project embodies a strong, real-time detection and recognition system of traffic signs that exhibits high precision rates and well performing in real time with a user interface created on streamlit. We obtained the dataset for training our model by carefully picking it out and preparing it through steps like binary thresholding, scaling and color space conversion. Our deep learning model which used TensorFlow and Keras achieved an incredible validation accuracy of 99.6% after 15 epochs (calculated using 1). A more engaging user experience is made possible by integrating live display with dynamic bounding boxes around detected signs as well as audible feedback via beeping sound upon recognition and sending notification alerts through SMS to the driver's phone number. This technology can have multiple applications such as traffic management systems or autonomous vehicles; thus, making road navigation safer and more efficient. By studying optimizations for handling complexities associated with complicated real-world scenarios, this may enhance future research towards enhancing its efficiency and applicability.

VII. Conflicts of interest

The authors declare no conflicts of interest

REFERENCES

[1] A. Avramović, D. Sluga, D. Tabernik, D. Skočaj, V. Stojnić and N. Ilc, "Neural-Network-Based Traffic Sign Detection and Recognition in High-Definition Images Using Region Focusing and Parallelization," in IEEE Access, vol. 8, pp. 189855-189868, 2020, doi: 10.1109/ACCESS.2020.3031191.

[2] M. Lopez-Montiel, U. Orozco-Rosas, M. Sánchez-Adame, K. Picos and O. H. M. Ross, "Evaluation Method of Deep Learning-Based Embedded Systems for Traffic Sign Detection," in IEEE Access, vol. 9, pp. 101217-101238, 2021, doi: 10.1109/ACCESS.2021.3097969.

[3] C. Dewi, R. -C. Chen, Y. -T. Liu, X. Jiang and K. D. Hartomo, "Yolo V4 for Advanced Traffic Sign Recognition With Synthetic Training Data Generated by Various GAN," in IEEE Access, vol. 9, pp. 97228- 97242, 2021, doi: 10.1109/ACCESS.2021.3094201

[4] H. Zhang et al., "Real-Time Detection Method for Small Traffic Signs Based on Yolov3," in IEEE Access, vol. 8, pp. 64145-64156, 2020, doi: 10.1109/ACCESS.2020.2984554.

[5] L. Wang, K. Zhou, A. Chu, G. Wang and L. Wang, "An Improved Light-Weight Traffic Sign Recognition Algorithm Based on YOLOv4- Tiny," in IEEE Access, vol. 9, pp. 124963-124971, 2021, doi: 10.1109/ACCESS.2021.3109798.

[6] D. C. Santos et al., "Real-Time Traffic Sign Detection and Recognition using CNN," in IEEE Latin America Transactions, vol. 18, no. 03, pp. 522-529, March 2020, doi: 10.1109/TLA.2020.9082723.

[7] M. D. RADU, I. M. COSTEA and V. A. STAN, "Automatic Traffic Sign Recognition Artificial Inteligence - Deep

Learning Algorithm," 2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Bucharest, Romania, 2020, pp. 1-4, doi: 10.1109/ECAI50035.2020.9223186.

[8] Triki, N.; Karray, M.; Ksantini, M. A Real-Time Traffic Sign Recognition Method Using a New AttentionBased Deep Convolutional Neural Network for Smart Vehicles. Appl. Sci. 2023, 13, 4793. https://doi.org/10.3390/app13084793

[9] et.al, A. V. (2021). Traffic Sign Classification Using Convolutional Neural Networks and Computer Vision. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(3), 4244–4250. Retrieved from https://www.turcomat.org/index.php/turkbilmat/article/view/1715

[10] E. Oruklu, D. Pesty, J. Neveux and J. -E. Guebey, "Real-time traffic sign detection and recognition for incar driver assistance systems," 2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS), Boise, ID, USA, 2012, pp. 976-979, doi: 10.1109/MWSCAS.2012.6292185.

[11] A. Vennelakanti, S. Shreya, R. Rajendran, D. Sarkar, D. Muddegowda and P. Hanagal, "Traffic Sign Detection and Recognition using a CNN Ensemble," 2019 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 2019, pp. 1-4, doi: 10.1109/ICCE.2019.8662019.

[12] R. M. Oza, A. Geisen and T. Wang, "Traffic Sign Detection and Recognition using Deep Learning," 2021 4th International Conference on Artificial Intelligence for Industries (AI4I), Laguna Hills, CA, USA, 2021, pp. 16-20, doi: 10.1109/AI4I51902.2021.00012.

[13] P. Dollar, C. Wojek, B. Schiele and P. Perona, "Pedestrian Detection: An Evaluation of the State of the Art," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 4, pp. 743-761, April 2012, doi: 10.1109/TPAMI.2011.155.

[14] Z. Zou, K. Chen, Z. Shi, Y. Guo and J. Ye, "Object Detection in 20 Years: A Survey," in Proceedings of the IEEE, vol. 111, no. 3, pp. 257- 276, March 2023, doi: 10.1109/JPROC.2023.3238524.

[15] Luca Parisi, Renfei Ma, Narrendar RaviChandran, Matteo Lanzillotta, hyper-sinh: An accurate and reliable function from shallow to deep learning in TensorFlow and Keras, Machine Learning with International Journal of Intelligent Systems and Applications in Engineering IJISAE, 202x, x(x), xx–xx | 7 Applications, Volume 6, 2021, 100112, ISSN 2666-827

[16] Patil, R. A. ., and P. D. . Patil. “Skewed Evolving Data Streams Classification With Actionable Knowledge Extraction Using Data Approximation and Adaptive Classification Framework”. International Journal on Recent and Innovation Trends in Computing and Communication, vol. 11, no. 1, Feb. 2023, pp. 38-52, doi:10.17762/ijritcc.v11i1.5985.

[17] Patil, R.A., Patil, P.D. Efficient approximation and privacy preservation algorithms for real time online evolving data streams. World Wide Web 27, 5 (2024). https://doi.org/10.1007/s11280-024-01244-9

[18] Ahire, P.G., Patil, P.D. CAC-WOA: context aware clustering with whale optimization algorithm for knowledge discovery from multidimensional space in electricity application. Cluster Comput 27, 499–513 (2024). https://doi.org/10.1007/s10586-023-03965-4

[19] Ahire, P. G. ., & Patil, P. D. . (2023). Context-Aware Clustering and the Optimized Whale Optimization Algorithm: An Effective Predictive Model for the Smart Grid. International Journal on Recent and Innovation Trends in Computing and Communication, 11(1), 62–76. https://doi.org/10.17762/ijritcc.v11i1.5987