Revolutionizing Traffic Sign And Signal Detection With Deep Learning Techniques

1st Hruthik Vinnakota
Department of Applied Data Science
San José State University
San Jose, USA
hruthik.vinnakota@sjsu.edu

2nd Parineeta Begrai
Department of Applied Data Science
San José State University
San Jose, USA
parineeta.begrai@sjsu.edu

3rd Nimisha Gupta
Department of Applied Data Science
San José State University
San Jose, USA
nimisha.gupta@sjsu.edu

Abstract—**Recognizing traffic signs has become more and more crucial in urban areas globally because of the increasing prevalence of advanced vehicles. With the widespread use of smart systems, having a dependable traffic sign recognition system is essential for ensuring safe driving and effective autonomous navigation. In Advanced Driver Assistance Systems (ADAS), the accurate identification of traffic signs plays a crucial role in improving both driver safety and comfort, particularly in difficult driving conditions such as adverse weather or unclear signage. This project aims to develop a novel deep learning framework designed specifically for precise recognition of traffic signs across different benchmark data sets, including those accessible on Kaggle, through the utilization of German Traffic Sign Recognition Benchmark image data sets. The proposed architecture combines residual convolutional blocks and hierarchical dilated skip connections to improve accuracy in identifying and classifying traffic signs. The primary objective is to highlight the vital role of traffic sign recognition in ADAS, ultimately enhancing driver safety and comfort, particularly in challenging road conditions. Challenges such as varying illumination, weather conditions, occlusion, deformation, and sign deterioration underscore the need for real-time solutions. Introducing a deep network with spatial transformer layers and modified inception modules, such as convolutional neural networks, Lenet, AlexNet, and VGG, suggests a promising solution for the classification of traffic signs in autonomous vehicles. This approach aims to address the complex nature of road traffic scenes and contribute to the development of effective, real-time solutions for traffic sign recognition.**

Index Terms—**Deep Learning, Image Processing, CNN, LeNet, AlexNet, VGG.**

I. Motivation

In the ever-evolving landscape of technological advancements, the profound impact on human life, particularly in the domain of transportation and traffic systems, cannot be overstated. The ongoing progress in technology has not only simplified daily lives but has also significantly contributed to the efficiency and safety of the transportation systems. The intriguing aspect lies in witnessing the continuous enhancements in technology that not only make our lives more convenient but also reshape the way by interacting with our surroundings. Among the technological marvels, Artificial Intelligence (AI), machine learning, and visual recognition stand out as transformative forces, particularly in the realm of autonomous driving technology. The integration of these cutting-edge technologies has propelled the development of systems capable of learning, adapting, and making informed decisions in real-time, revolutionizing the landscape of transportation. The primary motivation behind the project is rooted in harnessing the power of deep learning methods to address a critical aspect of autonomous driving – the accurate categorization of traffic rules. By developing a robust system capable of recognizing and categorizing traffic signs with precision, this research aims to empower autonomous vehicles to navigate roads seamlessly. The ultimate goal is to contribute to the creation of a driving experience that is not only efficient but also ensures the safety and comfort of individuals on the road. In essence, this Initiative seeks to be at the forefront of innovation, driving positive change in the way autonomous vehicles interact with and interpret the intricate web of traffic rules. Through a meticulous focus on AI-driven traffic sign recognition, determined to be a part of the transformative wave that is reshaping the future of transportation, making it safer, smarter, and more intuitive for everyone involved.

II. Background information

The foundation of the project work is firmly grounded in insights provided by authoritative sources, such as the National Highway Traffic Safety Administration (NHTSA), shedding light on the alarming frequency of road accidents attributed to the misinterpretation of traffic signs. According to data from the NHTSA, road accidents are recurrent, and a significant number can be traced back to errors in recognizing and understanding traffic signs. This underscores the critical need for advancements in traffic sign recognition systems to enhance road safety. Moreover, the main focus extends to human factors and errors associated with traffic sign recognition, drawing from research published in esteemed journals such as the Psychology of Transportation Research Journal. By delving into the nuances of human cognition and perceptual processes, we aim to gain a comprehensive understanding of the challenges individuals face in accurately interpreting traffic signs, contributing valuable insights to the development of effective recognition systems. To bolster our understanding, we analyze law enforcement records, tapping into data on fines and penalties issued for misreading traffic signs.

This meticulous investigation into legal consequences provides a tangible perspective on the real-world impact of misinterpretations. By sourcing information from local and national traffic databases, aim to quantify the financial and legal repercussions individuals face due to misreading traffic signs, emphasizing the pressing need for improved Road signal perception technologies. In parallel, this project aligns with the growing discourse surrounding the safe operation of autonomous vehicles. Reports from reputable companies and research institutions consistently highlight the paramount importance of precise highway signage identification for the successful integration of autonomous vehicles into our transportation systems. This recognition is crucial not only for the efficiency of autonomous vehicles but also for ensuring the safety of passengers and pedestrians alike. In substance, this research is strategically positioned at the intersection of comprehensive data insights, a profound understanding of human factors, and the contemporary discourse on autonomous vehicles. By synthesizing information from authoritative sources, the aim is to contribute to the development of innovative solutions that address the pressing challenges associated with Traffic indication identification, ultimately fostering safer and more efficient roadways.

III. Literature Review

The study reported by Shangyou Zen et al. (October 2022) provides a comprehensive document [1] delves into the meticulous process of data acquisition from two significant datasets: the German Traffic Sign Recognition Benchmark (GTSRB) and the Belgium Traffic Sign Dataset (BTSD). The models under scrutiny in this research include a conventional Convolutional Neural Network (CNN) and an enhanced CNN featuring a distinct Feature Extraction Module known as the TS-Module. Evaluation in the research is conducted through the use of two key metrics: accuracy and Caffe mode size (KB). The findings reveal that the improved CNN model, named MyNet, surpasses the performance of the traditional CNN model, Tra-Net, across both data sets. MyNet achieves higher accuracy rates of 97.4 percent and 98.1 percent, respectively, showcasing its superior recognition capabilities. Additionally, MyNet attains a smaller model size, as reflected in the Caffe mode size (KB), making it more portable and resource efficient. These results signify the effectiveness of MyNet in the domain of traffic sign recognition, highlighting its potential for practical application in real-world scenarios.

Titled ”Traffic sign classification using CNN and detection using faster-RCNN and YOLOV4” [2] this recent paper addresses previous techniques and algorithms for traffic sign recognition and object detection. Njayou Youssouf et al, (August 2022) collected and organized the dataset for training and testing from the German Traffic Sign Recognition Benchmark (GTSRB). They used a Convolutional Neural Network (CNN) for traffic sign classification, achieving a high accuracy of 99.20 percent with 0.8 million parameters. For traffic sign detection, the authors utilized the Faster R-CNN and YOLOv4 networks. The Faster R-CNN achieved a mean average precision (mAP) of 43.26 percent at 6 Frames Per Second (FPS), while YOLOv4 achieved an mAP of 59.88 percent at 35 FPS, making it the preferred model for real-time traffic sign detection. In the comparative analysis, YOLOv4 outperformed Faster R-CNN in terms of mAP and speed.

Latha Parameswaran et al, (March 2021) proposed deep learning architecture called capsule networks for traffic sign detection [3]. The data used in the study is the German Traffic Sign Recognition Benchmark (GTSRB) dataset, which consists of 51,840 images of 43 different classes. The dataset is divided into training data (39,209 images) and testing data (12,630 images). The model proposed in the paper is based on capsule networks, which address the limitations of convolutional neural networks (CNNs) in capturing the pose, view, and orientation of traffic sign images. Capsule networks use capsules, which are groups of neurons representing the parameters of an object like pose and orientation. Dynamic routing and route by agreement algorithms are used to compute the pose information of the capsules. The network architecture, preprocessing techniques, and loss functions used in the model are explained in detail. The evaluation metric used in the study is the correct classification rate (CCR), which represents the accuracy of the model in identifying traffic signs correctly. The model achieved an accuracy of 97.6 percent on the GTSRB dataset, outperforming other methods such as Random Forests and Linear Discriminant Analysis.

Jurisic et al. (2018) gathered data from multiple publicly available European traffic sign datas ets, including the German Traffic Sign Recognition Benchmark (GTSRB), the Belgian Traffic Sign Dataset (BTSD), and others. They used a Convolutional Neural Network (CNN) with Spatial Transformer Networks (STN) for their traffic sign recognition system [4]. The evaluation metric used in the paper is accuracy, and the results are presented in the form of comparison tables and figures. The proposed CNN with 3 STNs achieved an accuracy of 98.87 percent on the GTSRB dataset, outperforming existing methods such as GDBM, OneCNN, and INNLP + SRC(PI). On the same data set, their CNN with 3 STNs achieved an accuracy of 99.71 percent, making it the top-ranked model. The proposed CNN also demonstrated lower memory requirements and a lower number of parameters compared to existing methods.

IV. Methodology

A. Data Collection

The dataset is sourced from the German Traffic Sign Recognition Benchmark comprising more than 50,000 images across more than 40 classes, facilitating a realistic and challenging scenario for model training and evaluation. This dataset provides classes meta information about classes which include-path, classID, shapeID, colorID, and signID. These details help in understanding the characteristics of each traffic sign category.

Both train and test packages include functions for reading ground truth data, evaluating self-implemented detector functions by presenting downloaded images individually, as well as assessing a text file containing the results of the detector. The performance is further evaluated on the test dataset.

B. Data Exploration

We employed a Python script to explore the dataset, revealing that the training set comprises 43 folders, each dedicated to traffic sign images of a specific class. Consequently, there are a total of 43 distinct traffic signs represented in Figure 1. In the test set, we encountered a sum of 12,631 images. Upon utilizing another Python script to generate a bar chart illustrating the distribution of images for each label, a noteworthy observation emerged: the training dataset exhibits an imbalance. This imbalance underscores the necessity to rectify the dataset's distribution before proceeding with modeling. Furthermore, during our examination, we noted the presence of both blurred and dark images in both the training and test folders. This observation underscores the importance of testing our model on such dark images to assess its performance under challenging conditions.

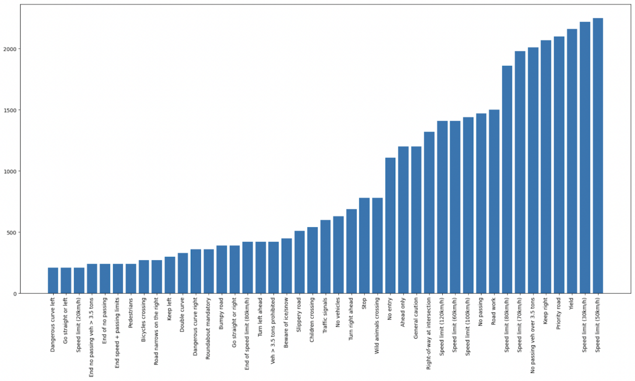


Fig. 1. Exploration process

C. Data Preprocessing

To prepare the data for modeling, we resize the images to match the input shape of the model. Additionally, we scaled the images in both the training and validation set to a range between 0 and 1. This scaling facilitates faster computations during the modeling process. The traffic sign labels were converted into numeric values using one-hot encoding. Moreover, to enhance the diversity of our training data and improve the model's generalization, we implemented data augmentation techniques. This involves creating additional training samples by applying transformations such as rotation and cropping to the original images. This augmented dataset helps the model learn robust features and patterns, leading to better performance during training and testing. To ensure that the labels are preserved after augmenting, we have used the 'ImageDataGenerator' class from the TensorFlow library.

D. Data Preparation

The training of our model is based on traffic sign images from the train folder. To ensure the model's effectiveness on the test folder images, we divided the images in the train folder into a training set and a validation set in an 80:20 ratio. This allows us to evaluate the model's performance on the validation set before assessing its final performance on the test set.

E. Model Architecture

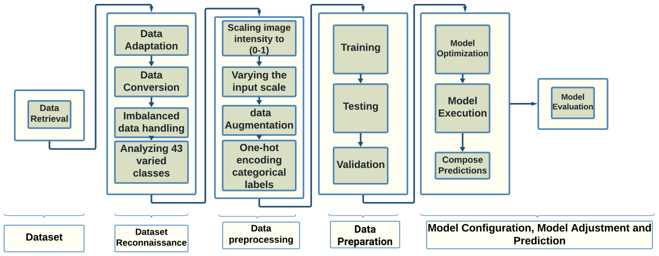


Fig. 2. Model workflow

As depicted in Figure 2, our project follows a structured workflow. We begin by collecting data from Kaggle, specifically utilizing the (GTSR) Benchmark image datasets. Subsequently, we conduct data exploration, including adaptation, conversion, handling imbalanced data, and analyzing 43 different classes. Subsequent to this, we preprocess the data by scaling image intensity to a range of 0 to 1, adjusting the input scale, and implementing data augmentation. Crucially, we use one-hot encoding for categorical labels. After thorough data preparation, Following meticulous data preparation, introduced hyperparameter tuning techniques before advancing to the training and validation phases, utilizing the refined data. followed by testing using separate test images. The modeling phase involves optimization, execution, and prediction composition. Finally, after model evaluation and generated the results accordingly.

V. Models

In this project, our goal is to categorize traffic sign images into 43 distinct labels. To achieve this, we are implementing four different models: Convolutional Neural Network (CNN), LeNet, AlexNet, and a pre-trained VGG model. To address the issue of imbalanced data, where some classes have more examples than others, we are utilizing class weights. This ensures that the model does not become biased toward the majority class, promoting a more equitable learning process across all classes. Preserving the color information of traffic sign images is crucial for our task, so we are intentionally avoiding grayscale conversion during training. Recognizing the significance of color in traffic sign images, we aim to train our models without compromising this valuable aspect.

- **Model 1:** The VGG model, which stands for Visual Geometry Group, has been implemented as the first model for our project. This model, also known as VGG16, has 16 layers. Its hallmark simplicity lies in the strategic stacking of compact 3x3 convolutional filters across multiple layers, culminating in a deep and uniform structure. This design renders VGG particularly effective in image classification tasks. Within the German Traffic Sign Recognition Benchmark, VGG has demonstrated a remarkable ability to learn hierarchical representations of traffic signs, proving instrumental in discerning nuanced details crucial for precise classification. Its proficiency extends to the identification and categorization of traffic signs, positioning VGG as a valuable asset in advancing road safety through enhanced recognition capabilities.

For our project's image classification task, we opted for the VGG16 model as the foundation. This choice stemmed from its demonstrated effectiveness across various computer vision tasks, aligning well with the objectives of our project. Building upon the VGG16 base model, we introduced a Global Average Pooling layer followed by a dense layer. The Global Average Pooling layer proved instrumental in diminishing the spatial dimensions of the base model's feature maps, effectively retaining crucial information for classification. Meanwhile, the subsequent dense layer served the purpose of generating the requisite number of classes for our specific classification task. This combination of layers empowered us to construct a resilient and accurate image classification model.

- **Model 2:** Convolutional Neural Networks (CNNs) are a commonly used architecture for image classification tasks. They perform exceptionally well in recognizing patterns and features in images. CNNs have multiple convolutional layers that can learn hierarchical features from input images, making them ideal for complex image recognition tasks. In traffic sign recognition, CNNs are particularly effective at capturing spatial hierarchies and patterns within the signs. This allows for more accurate and robust classification, even when the signs are partially occluded or damaged. To further improve the performance of CNNs in traffic sign recognition, researchers often use techniques such as pooling and dropout. Pooling involves reducing the dimensionality of feature maps, which can enhance computational efficiency and reduce overfitting. Dropout randomly drops out some of the nodes in the network during training, which helps the network learn more robust and generalizable features. By using these techniques, CNNs can improve their accuracy on the training set and enhance their ability to generalize across various traffic sign instances. This leads to better performance on new, unseen data.

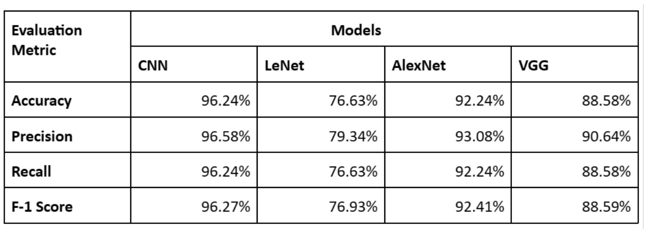
The custom CNN model is constructed using a sequential architecture in TensorFlow. It comprises three convolutional layers with 32, 64, and 128 filters, respectively, each followed by max-pooling layers with a 2x2 pool size. The rectified linear unit (ReLU) activation function is applied to introduce non-linearity in the convolutional layers. Subsequently, a flattening layer transforms the output into a 1D array, followed by a densely connected layer with 128 units and ReLU activation. To mitigate overfitting, a dropout layer is incorporated, with the dropout rate being a hyperparameter optimized during tuning. The final layer employs the softmax activation function for multi-class classification. The model is compiled using the Adam optimizer with a learning rate and L2 decay as hyperparameters. The categorical cross-entropy loss function is chosen for training, and accuracy is selected as the evaluation metric.

- **Model 3:** LeNet-5 stands as one of the pioneering convolutional neural network (CNN) architectures designed for image classification, specifically for recognizing handwritten and machine-printed characters. With a straightforward architecture comprising five layers, including three sets of convolution layers interspersed with average pooling, LeNet-5 demonstrated its effectiveness in feature extraction and classification tasks. The model processes 32x32 grayscale images through successive convolution and pooling operations, using activation function as tanh, ultimately leading to fully connected layers and a Softmax classifier for classification into distinct classes. In this project, we have implemented a modified version of LeNet architecture that consists of 32x32 colored images and 'relu' as the activation function. The model commences with a convolutional layer with 6 filters of size 5x5, followed by average pooling, reducing the feature map dimensions. Subsequently, a second convolutional layer with 16 filters is employed, again followed by average pooling. The flattened output is then connected to two fully connected layers of 120 and 84 neurons, respectively, before culminating in an output layer with 43 neurons for classification. The total trainable parameters amount to 85,931, highlighting the model's capacity to adapt and learn from the given dataset. The hyperparameters utilized in training involve 50 epochs, a batch size of 128, and class weights incorporated for addressing class imbalance. The training also incorporates early stopping with patience of 5 epochs to prevent overfitting. This LeNet architecture, along with the specified hyperparameters, demonstrates a comprehensive approach to image classification, emphasizing its adaptability and efficiency in handling diverse datasets.

- **Model 4:** AlexNet, It comprises eight layers including five convolutional layers followed by max-pooling layers and three fully connected layers. Dropout layers are strategically placed to combat overfitting. Rectified Linear Unit (ReLU) activation functions is introduced, bringing about faster convergence during training.AlexNet is deeper and more complex than LeNet, with more convolutional and fully connected layers, allowing it to capture intricate features in images.LeNet primarily used sigmoid and tanh activation functions, while AlexNet introduced the more computationally efficient and faster converging ReLU activation functions. The network's architecture implemented in this project is as follows: two initial convolutional layers with max-pooling, followed by three more convolutional layers, another max-pooling layer, and concluding with three fully connected layers. The convolutional layers use rectified linear unit (ReLU) activation functions, and dropout is applied to the fully connected layers for regularization. The hyperparameters and parameters include a learning rate, batch size, dropout rate, and the number of epochs during training. The learning rate determines the step size during optimization (set to 0.0001), while the batch size specifies the number of training samples processed in a single iteration (128). Dropout (0.5) is applied to fully connected layers, which aids in preventing overfitting. The number of epochs is set as 30. The model's total parameters amount to 24,903,083, which is higher than LeNet, contributing to its large capacity for learning intricate features in data. These hyperparameters and parameters collectively govern the training and performance of AlexNet.

VI. Experiments and Results

Our project focuses on image classification, and for evaluation, we employ key metrics such as accuracy, precision, recall, and F-1 score in Figure 3. Accuracy reflects the total correct predictions, while precision measures the ratio of true positive predictions to the total predicted positives. Recall, on the other hand, assesses the ratio of true positive predictions to the total actual positives. The F1 score, computed as the harmonic mean of precision and recall, is a comprehensive metric. Weighted averaging is chosen for computation due to the dataset's imbalance, featuring 43 class labels. This approach ensures fair consideration for each class, preventing metric bias towards the majority class.



Evaluation Metric	Models			
	CNN	LeNet	AlexNet	VGG
Accuracy	96.24%	76.63%	92.24%	88.58%
Precision	96.58%	79.34%	93.08%	90.64%
Recall	96.24%	76.63%	92.24%	88.58%
F-1 Score	96.27%	76.93%	92.41%	88.59%

Fig. 3. Evaluation metric results.

The VGG model is trained on the dataset with tuned hyperparameters, employing early stopping to mitigate overfitting. The training history, encompassing accuracy and loss, is monitored and visualized. Evaluation of the test dataset demonstrates notable performance in the German traffic sign classification task, achieving an accuracy of 88.58%, precision of 90.64%, recall of 88.58%, and an F1 score of 88.59%. The graph provides in Figure 4 a clear observation of both training accuracy and validation accuracy derived from the accuracy metric, as well as training loss and validation loss obtained from the loss metric.

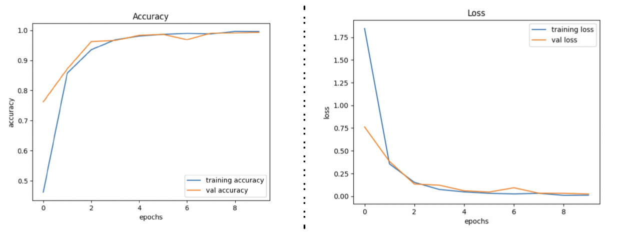


Fig. 4. Accuracy and Loss metric from VGG model

The resulting best CNN model achieves a test accuracy of approximately 96.25%, with precision, recall, and F1 score reported as 96.58%, 96.25%, and 96.27%, respectively.

From above graphs compared to the VGG model, our trained CNN model performs really well, especially during testing. When we put it to the test, the CNN model shows effectiveness and potential. It seems to do better than the VGG model, indicating that the CNN approach could be more reliable and effective, which is great for the specific testing needs we have.

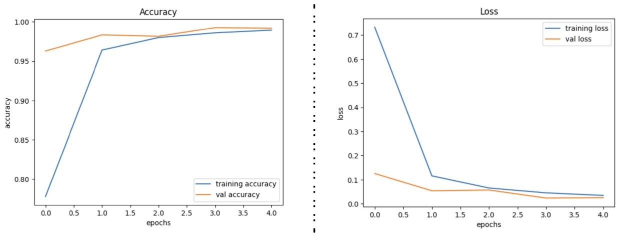


Fig. 5. Accuracy and Loss metric from CNN model

The outcomes of each model for the testing dataset are reported. The LeNet classification reports are displayed in Figure 6. With an accuracy rate of 76%, evaluation makes it clear that recall of 76.6% and precision 76.9%. In the testing phase are promising. But the CNN outperformed LeNet, exhibiting superior results. Specifically, the F1 values in LeNet 76.91% with all standard results.

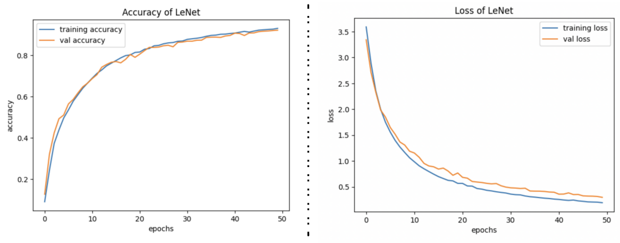


Fig. 6. Accuracy and Loss metric from LetNet model

Analyzing the output results of AlexNet from Figure 7 reveals precision, recall, and F1 scores all at 92% when using the testing data set. However, it's worth noting that other parameters, such as accuracy, are experiencing a decrease, with a value of 92.24%.

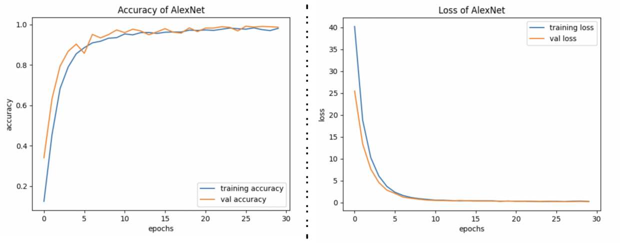
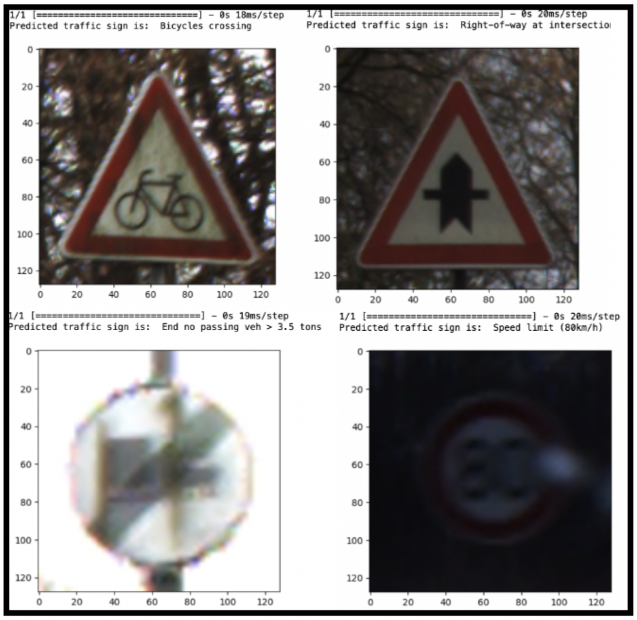


Fig. 7. Accuracy and Loss metric from AlexNet model

Upon comparing these metrics across different models, the CNN model outperformed others, achieving the highest accuracy score of 96.24%. The accompanying figure 4,5,6,7 illustrates the training accuracy vs validation accuracy, as well as training loss vs validation loss for all models. Tuning the models with diverse hyperparameter combinations resulted in a discernible improvement pattern over epochs, with both training and validation accuracies consistently rising. This pattern indicates the model's successful learning and generalization to the validation set, supported by a steady decline in the loss function. By thorough comparisons, identified CNN as our best-performing model. Further investigation revealed its capability to predict unseen images accurately from test sets, even detecting blurred or dark images, as demonstrated in Figure 8.



```
1/1 [==============================] - 0s 18ms/step
Predicted traffic sign is:  Bicycles crossing
```

```
1/1 [==============================] - 0s 20ms/step
Predicted traffic sign is:  Right-of-way at intersectio
```

```
1/1 [==============================] - 0s 19ms/step
Predicted traffic sign is:  End no passing veh > 3.5 tons
```

```
1/1 [==============================] - 0s 20ms/step
Predicted traffic sign is:  Speed limit (80km/h)
```

Fig. 8. Images from the test set predicted successfully

VII. Acknowledgment

We are deeply thankful to Professor Shayan Shams for his consistent guidance and valuable support, which significantly contributed to the successful completion of our project. Professor Shams shared his extensive expertise throughout the coursework and provided unwavering assistance. His dedication to our learning experience was exceptional and greatly enhanced our educational journey.

VIII. Conclusion

In the final analysis, our thorough comprehensive exploration of a variety of models has shown how effective they are in forecasting several kinds of traffic signs when measured against different standards. Each model had distinct benefits demonstrating the flexibility and adaptability of modern methods for classifying signboards. In the end, our experiment shows that sophisticated deep learning models are being used to identify street signs and highlights the potential of these aspects to be key components of innovative innovations that seek to revolutionize how people interact with and navigate traffic systems. The journey from diverse model exploration to the exceptional performance of the CNN encapsulates a significant stride towards the realization of safer, smarter, and more efficient transportation systems.

IX. Discussion and Future Improvement

Looking ahead, our project is geared towards continuous improvement, focusing on elevating the performance of our traffic sign recognition model. To achieve this, we plan to augment our dataset with a broader range of traffic sign data, enhancing the model's adaptability to challenging conditions. Additionally, we aim to integrate image segmentation, providing a more detailed understanding of the spatial context around each sign for improved accuracy. The integration of diverse data, image segmentation, and advanced model assembling or CNN variations will propel our project into a new realm of accuracy and efficiency, reinforcing its potential impact on the broader realm of transportation safety.

In our devised methodology, the evaluation of classification accuracy on the GTSRB dataset utilizing the CNN model yielded a commendable 96% accuracy. The strategic decision to retain color information played a pivotal role in achieving this notable accuracy. Additionally, the integration of class weights mitigated challenges associated with imbalanced datasets, ensuring a balanced and effective model.

The model showcased exceptional precision, recall, and F-1 scores for each class, indicative of its proficiency in minimizing both false positives and false negatives. Noteworthy is the model's adeptness in handling challenging scenarios, and accurately classifying images characterized by low visibility, blurriness, or rotation. Furthermore, its robust performance on previously unseen data, sourced from the internet, highlights its adaptability and generalization capabilities. However, there exist instances where the model encounters challenges, particularly when faced with significant alterations in color, shape, or orientation.

References

[1] Li, W., Li, D., & Zeng, S. (2019). Traffic Sign Recognition with a small convolutional neural network. IOP Conference Series, 688(4), 044034.

[2] Youssouf, N. (2022). Traffic sign classification using CNN and detection using faster-RCNN and YOLOV4. Heliyon, 8(12), e11792.

[3] Kumar, A. D. (2018). Novel deep learning model for traffic sign detection using capsule networks. arXiv (Cornell University).

[4] Arcos-García, Á., Álvarez-García, J. A., and Morillo, L. M. S. (2018). Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods. Neural Networks, 99, 158–165.

[5] Postovan, A., & Eraşcu, M. (2023). Architecturing binarized neural networks for traffic sign recognition. arXiv (Cornell University).

[6] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). 2012 AlexNet. Adv. Neural Inf. Process. Syst, 1-9.

[7] Kaggle dataset: https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign.