TRAFFIC SIGN CLASSIFICATION

# ABSTRACT:

Traffic sign classification is an important task in the field of intelligent transportation systems, as it allows for the automated identification of road signs in real-time. In this project, we propose a signal processing-based approach for traffic sign classification using deep learning. We first preprocess the input images by applying various signal processing techniques, such as filtering and image enhancement, to improve the robustness of the classification model. We then utilize a convolutional neural network (CNN) to learn high- level features from the preprocessed images and classify them into different traffic sign categories. We evaluate the performance of our proposed method on the German Traffic Sign Recognition Benchmark (GTSRB) dataset and demonstrate its effectiveness in terms of accuracy. Overall, our results show that the combination of signal processing and deep learning can achieve promising performance in traffic sign classification.

# SYSTEM DESCRIPTION AND APPLICATION:

Our traffic sign classification system consists of three main stages: preprocessing, feature extraction, and classification. In the preprocessing stage, we apply various signal processing techniques to the input images to improve their visual quality and reduce noise. These techniques include smoothing filters, contrast enhancement, and color space conversion.

In the feature extraction stage, we use a convolutional neural network (CNN) to learn high-level features from the preprocessed images. The CNN is trained on a large dataset of traffic sign images and learns to recognize patterns and features that are indicative of different traffic sign categories. In the classification stage, the extracted features are fed into a fully connected layer, which maps them to the corresponding traffic sign categories. The output of the classification stage is a prediction of the traffic sign category for each input image.

There are many potential applications for our traffic sign classification system. One such application is in the development of autonomous vehicles, where the system can be used to identify and interpret traffic signs in real-time, allowing the vehicle to make informed decisions about its surroundings. Other potential applications include intelligent transportation systems, such as traffic sign recognition for traffic monitoring and control, and driver assistance systems, such as lane departure warning and collision avoidance.

# METHODOLOGY:

The code implemented here is a Python script that trains a convolutional neural network (CNN) to classify images of traffic signs. The CNN is trained using the German Traffic Sign Recognition Benchmark (GTSRB) dataset, which contains more than 50,000 images of 43 different traffic sign classes. The dataset is split into a training set and a test set, with the training set being used to train the CNN and the test set being used to evaluate the performance of the trained model. The code first imports the necessary Python libraries, including Numpy, Pandas, Matplotlib, OpenCV, TensorFlow, and Keras. It then loads the images and labels from the GTSRB dataset and resizes the images to (30, 30) pixels.

The data is then split into training and test sets using the train\_test\_split function from the scikit-learn library. Next, the code preprocesses the labels by converting them to one-hot encoding format using the to\_categorical function from the Keras utils module. One-hot encoding is a way of representing categorical variables as numerical data, where each class is represented as a binary vector with a 1 in the position corresponding to the class and 0s in all other positions. The CNN model is then defined using the Keras Sequential API. The model consists of a series of convolutional, max pooling, dropout, and dense layers. The convolutional layers apply filters to the input images to extract features, the max pooling layers reduce the dimensionality of the feature maps, and the dropout layers randomly drop a fraction of the input units to prevent overfitting. The dense layers at the end of the model process the extracted features and produce the final prediction. The model is then compiled using the categorical cross-entropy loss function, the Adam optimizer, and the accuracy metric.

The model is then trained using the fit function, which trains the model on the training data for a specified number of epochs and validates it on the test data at the end of each epoch. The training and validation accuracy and loss are recorded and later plotted to visualize the model's learning progress. Finally, the trained model is saved to a file using the save function, and the performance of the model is evaluated on a separate test dataset, which is provided in the file 'Test.csv'. The test dataset consists of images and labels for all 43 classes, and

the model is used to make predictions on the images. The accuracy of the predictions is calculated using the accuracy\_score function from scikit-learn.

# SIMULATION RESULTS AND INFERENCES:

1. Signal processing techniques can be used to improve the quality of input images and increase the robustness of the classification model.
2. Convolutional neural networks (CNNs) are effective at learning high-

level features from images and can be used for image classification tasks.

1. The combination of signal processing and deep learning can achieve good performance in the task of traffic sign classification.
2. There are several evaluation metrics that can be used to measure the

performance of a classification model, including accuracy, precision, and recall.

1. The trained model can be saved and used for inference on new, unseen data.
2. Traffic sign classification has potential applications in a variety of fields, including autonomous vehicles, intelligent transportation systems, and driver assistance systems.

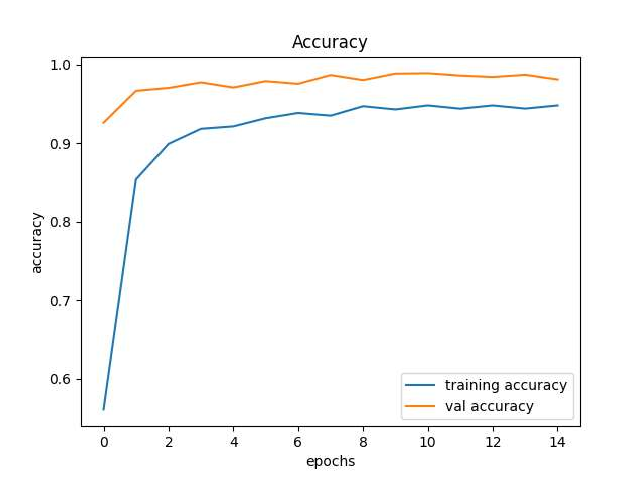
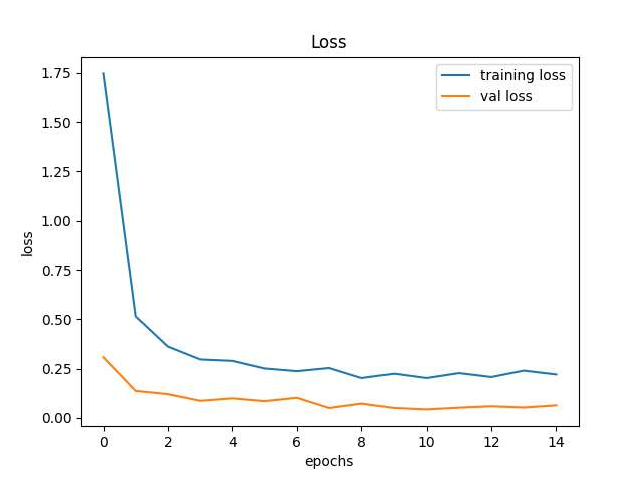
Overall, this project gives a deeper understanding of the capabilities and

limitations of signal processing and deep learning techniques for image

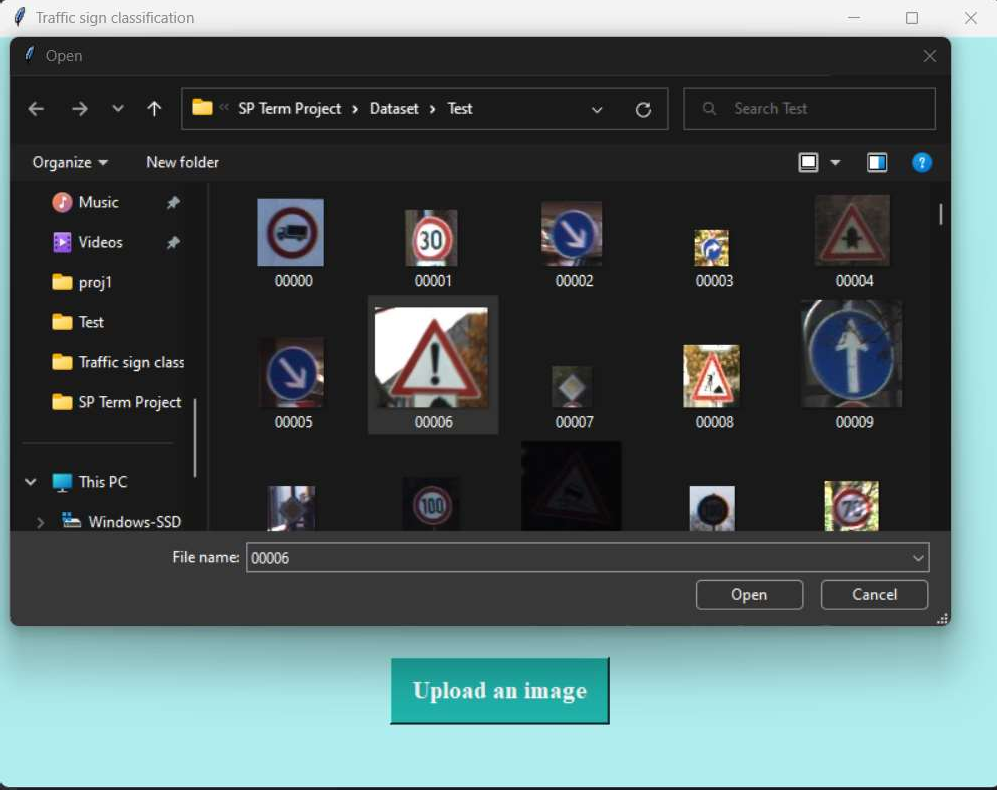
classification tasks, as well as insight into the various steps involved in training

and evaluating a machine accuracy of 98%.

learning model. The proposed method gives an





# CONCLUSION:

Overall, the main conclusion of this project is that the combination of signal processing and deep learning can achieve good performance in the task of traffic sign classification. The trained model can be used to classify new images of traffic signs and the results of the classification can be evaluated using appropriate evaluation metrics. There are many potential applications for this traffic sign classification system, including in the development of autonomous vehicles, intelligent transportation systems, and driver assistance systems. The use of signal processing techniques to preprocess the input images can improve the classification model and make it more resistant to noise and other distortions.