Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science engineering and Information Systems (SCORE)

FINAL PROJECT REPORT

(Artificial Intelligence-SWE4010)

WINTER SEMESTER 2023-2024

TRAFFIC SIGN DETECTION SYSTEM

Under the guidance of -GANESAN K

SUBMITTED BY:
ANNAPOORNA A NAIR-21MIS0130
ADVAIT MADHU-21MIS0034

ABSTRACT

The challenges of urban traffic management necessitate innovative solutions to enhance road safety and efficiency. This project presents a rigorous exploration into the development of an AI-based Traffic Sign Detection System, leveraging the VGG-16 algorithm. Through meticulous data acquisition, preprocessing, and model design, the system demonstrates robust capabilities in real-time detection and classification of diverse traffic signs.

The project commences with the acquisition of a meticulously curated dataset comprising annotated images of various traffic signs, ensuring representation across regulatory, warning, and informational categories. Emphasizing diversity in lighting conditions, weather variations, and environmental contexts enriches the dataset's resilience and generalizability. Subsequent data preprocessing techniques, including standardization of image resolutions, normalization of pixel values, and augmentation through transformations, establish a uniform and enriched dataset primed for model training.

Central to the system's architecture is the utilization of the VGG-16 convolutional neural network. Renowned for its depth and efficacy in image recognition tasks, VGG-16 embodies a hierarchical structure comprising multiple convolutional layers interspersed with max-pooling layers, culminating in fully connected layers for feature extraction and classification. The inherent depth of VGG-16 enables the system to discern intricate patterns and features crucial for accurate sign identification across diverse contexts.

The training process entails meticulous optimization of model parameters through gradient descent algorithms, minimizing classification error to ensure high precision and recall rates. Rigorous validation procedures, including cross-validation techniques and performance metrics evaluation, attest to the system's robustness and reliability across varying traffic sign scenarios.

Empirical evaluations underscore the system's efficacy, exhibiting high accuracy rates in real-time sign detection and classification. The scalability of the system, facilitated by VGG-16's architectural versatility, underscores its applicability to diverse traffic management contexts. Furthermore, the system's computational efficiency ensures feasibility for deployment in resource-constrained environments, fostering its practical utility in real-world traffic scenarios.

This project underscores the pivotal role of advanced machine learning techniques, exemplified by the VGG-16 algorithm, in addressing contemporary challenges in traffic management. By combining rigorous data preprocessing, model optimization, and empirical validation, the AI-based Traffic Sign Detection System presented herein offers a compelling framework for augmenting road safety measures and optimizing traffic management strategies in urban environments.

PROBLEM STATEMENT

The management of urban traffic poses multifaceted challenges, compounded by escalating vehicular densities, evolving road infrastructure, and dynamic regulatory requirements. Manual monitoring and enforcement of traffic regulations, particularly pertaining to the identification and interpretation of traffic signs, often prove laborious, error-prone, and susceptible to human limitations. In this context, the absence of efficient, automated systems for traffic sign detection and classification undermines road safety measures and exacerbates the risk of vehicular accidents and traffic congestion.

Traditional methods of traffic sign detection predominantly rely on manual surveillance and human intervention, which are inherently limited by factors such as human fatigue, cognitive biases, and environmental conditions. The reliance on human operators for real-time sign identification introduces delays, inefficiencies, and inaccuracies in traffic management processes, impeding the seamless flow of vehicular and pedestrian traffic. Moreover, the sheer diversity and complexity of traffic sign variations, encompassing regulatory, warning, and informational categories, further exacerbate the challenges associated with manual sign interpretation and enforcement.

The inadequacy of existing traffic management paradigms underscores the imperative for innovative, technology-driven solutions to augment road safety measures and streamline traffic management operations. Specifically, the development of an AI-based Traffic Sign Detection System emerges as a compelling approach to mitigate the shortcomings of traditional methods and foster a paradigm shift towards automated, data-driven traffic management strategies.

The key challenges addressed by the proposed Traffic Sign Detection System encompass the following:

1. **Real-time Detection:** The system must exhibit real-time capabilities to detect and classify traffic signs swiftly and accurately, facilitating prompt response and decision-making in dynamic traffic scenarios.

2. **Robustness to Variability:** The system should demonstrate resilience to variability in environmental conditions, lighting conditions, weather variations, and spatial orientations, ensuring consistent performance across diverse operational contexts.

3. **Multi-class Classification:** Given the diverse typologies of traffic signs, encompassing regulatory, warning, and informational categories, the system must support multi-class classification to accurately discern and interpret different sign types.

4. **Scalability and Generalizability:** The system's architecture should be scalable and adaptable to accommodate evolving traffic sign variations, infrastructural changes, and regulatory updates, ensuring long-term viability and relevance.

5. **Computational Efficiency:** Considering the exigencies of real-world traffic management scenarios, the system should prioritize computational efficiency, minimizing processing time and resource utilization without compromising accuracy or reliability.

Addressing these challenges necessitates a comprehensive approach encompassing data acquisition, preprocessing, model development, and validation, guided by principles of machine learning, computer vision, and deep learning. By developing an AI-based Traffic Sign Detection System capable of seamlessly integrating into existing traffic management infrastructures, this project endeavours to enhance road safety measures, optimize traffic management strategies, and foster sustainable urban mobility.

DATASET DESCRIPTION

The foundation of any machine learning-based system lies in the quality and diversity of its training data. In the context of the AI-based Traffic Sign Detection System, the dataset plays a pivotal role in facilitating model training, validation, and evaluation. The dataset encompasses a comprehensive collection of annotated images depicting various traffic signs encountered in real-world scenarios, encompassing regulatory, warning, and informational categories.

Data Acquisition: The dataset acquisition process involves sourcing annotated images from diverse sources, including publicly available repositories, traffic sign databases, and proprietary datasets. Emphasis is placed on ensuring representation across different geographical regions, road types, and environmental contexts to capture the inherent variability in traffic sign appearances. Collaboration with transportation authorities, research institutions, and urban planning agencies may facilitate access to specialized datasets tailored to specific geographic regions or regulatory regimes.

Dataset Composition: The dataset comprises a heterogeneous mix of traffic signs, encompassing a wide range of shapes, colours, sizes, and textual annotations. Sign categories include but are not limited to:

1. Regulatory Signs: Signs conveying mandatory instructions, prohibitions, and restrictions, such as speed limits, stop signs, and no-entry signs.

2. Warning Signs: Signs indicating potential hazards, road conditions, or changes in traffic patterns, including pedestrian crossings, slippery road warnings, and animal crossing signs.

3. Informational Signs: Signs providing guidance, directions, or information to motorists and pedestrians, such as route markers, destination indicators, and service area signage.

Annotated Ground Truth: Each image in the dataset is meticulously annotated with bounding boxes or segmentation masks delineating the spatial extent of individual traffic signs. Additionally, textual labels corresponding to the sign's semantic class (e.g., "speed limit 30 km/h", "yield") provide ground truth annotations essential for supervised learning tasks.

Data Preprocessing: Preprocessing techniques are applied to standardize and augment the dataset, enhancing model robustness and generalizability. Common preprocessing steps include:

1. Image Resizing: Standardizing image resolutions to a uniform size ensures consistency in input dimensions across the dataset.

2. Pixel Normalization: Normalizing pixel values to a common scale mitigates illumination variations and enhances model convergence during training.

3. Data Augmentation: Augmenting the dataset through geometric transformations (e.g., rotation, translation, flipping) introduces variability and diversifies the training samples, improving model generalization capabilities.

Dataset Splitting: The dataset is partitioned into distinct subsets for training, validation, and testing purposes, ensuring independent evaluation of model performance and generalization to unseen data. Stratified sampling techniques may be employed to preserve class distributions across different subsets, mitigating biases and ensuring representative evaluation metrics.

Dataset Characteristics:

- **Size:** The dataset comprises a substantial number of images, depending on the scope and scale of the project.

- **Diversity:** The dataset exhibits diversity in traffic sign types, environmental conditions, lighting variations, and spatial orientations, reflecting real-world traffic scenarios.

- **Annotations:** Ground truth annotations provide detailed spatial and semantic information essential for model training and evaluation, facilitating precise localization and classification of traffic signs.

CODE:

RESIZING DATA SET TO A COMMON RATIO.

```
import cv2
import os
import numpy as np

# Set the path to your dataset
dataset_path = "traffic_sign_classification_dataset\\train"

# Parameters
input_shape = (224, 224)


# Function to resize and preprocess images in the dataset
def preprocess_dataset(dataset_path, input_shape):
    for folder_name in os.listdir(dataset_path):
        folder_path = os.path.join(dataset_path, folder_name)
        for filename in os.listdir(folder_path):
            image_path = os.path.join(folder_path, filename)
            image = cv2.imread(image_path)

            # Resize image
            image_resized = cv2.resize(image, input_shape)

            # Perform additional preprocessing steps, such as normalization
            image_normalized = image_resized / 255.0

            # Save the resized and preprocessed image back to the dataset folder
            cv2.imwrite(
                image_path, (image_normalized * 255).astype(np.uint8)
            )  # Convert back to 0-255 range and save


# Preprocess the dataset
preprocess_dataset(dataset_path, input_shape)
```

MODEL TRAINING

Model training constitutes a pivotal phase in the development of the AI-based Traffic Sign Detection System, wherein machine learning algorithms are iteratively refined and optimized to discern patterns and features essential for accurate sign detection and classification. This section elucidates the intricacies of the training process, encompassing dataset preparation, model architecture design, optimization techniques, and performance evaluation protocols.

Dataset Preparation: Before commencing model training, the dataset undergoes meticulous preprocessing to standardize input data and augment training samples, enhancing model robustness and generalization capabilities. Preprocessing steps include image resizing, pixel normalization, and data augmentation through geometric transformations to diversify training samples and mitigate overfitting.

Model Architecture Design: The choice of model architecture profoundly influences the system's performance and computational efficiency. In the context of traffic sign detection, convolutional neural networks (CNNs) emerge as a prevalent choice due to their efficacy in image recognition tasks. The VGG-16 architecture, characterized by its deep hierarchical structure comprising multiple convolutional and pooling layers, offers a compelling framework for extracting intricate features crucial for sign detection and classification.

Training Procedure: The training procedure involves iteratively optimizing model parameters to minimize classification error and enhance predictive accuracy. Key components of the training process include:

1. **Loss Function Selection:** A suitable loss function, such as categorical cross-entropy, is chosen to quantify the disparity between predicted and ground truth labels during model training. The choice of loss function influences the model's ability to converge towards optimal parameter values and optimize classification performance.

2. **Gradient Descent Optimization:** Model parameters are updated iteratively using gradient descent optimization techniques, such as stochastic gradient descent (SGD) or adaptive optimization algorithms like Adam. The gradients of the loss function with respect to model parameters guide parameter updates, steering the model towards regions of lower loss and improved performance.

3. **Hyperparameter Tuning:** Hyperparameters, including learning rate, batch size, and regularization parameters, are fine-tuned through systematic experimentation to optimize model convergence and prevent overfitting. Cross-validation techniques may be employed to assess model generalization across different dataset partitions and hyperparameter configurations.

4. **Backpropagation:** The backpropagation algorithm facilitates efficient computation of gradient updates by propagating error gradients backward through the network, enabling parameter adjustments in proportion to their contribution to the overall loss. Through successive iterations of forward and backward passes, the model iteratively learns to better discriminate between different traffic sign classes.

MODEL TRAINING CODE

```
import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
import os
from tqdm import tqdm
import cv2
import numpy as np

# Set the path to your dataset
dataset_path = "traffic_sign_classification_dataset\\train"

# Parameters
input_shape = (224, 224, 3)  # Adjusted input shape to include channel dimension
num_classes = len(os.listdir(dataset_path))
epochs = 10  # Specify the number of epochs

# Data Augmentation
train_datagen = ImageDataGenerator(
    rescale=1.0 / 255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    validation_split=0.2,
)

# Split the dataset into training and validation sets
train_generator = train_datagen.flow_from_directory(
    dataset_path,
    target_size=(224, 224),  # Resize images to (224, 224)
    batch_size=32,
    class_mode="categorical",
    subset="training",
)

val_generator = train_datagen.flow_from_directory(
```

```python
    dataset_path,
    target_size=(224, 224),  # Resize images to (224, 224)
    batch_size=32,
    class_mode="categorical",
    subset="validation",
)

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense


def create_vgg19_model(input_shape, num_classes):
    model = Sequential()
    model.add(
        Conv2D(64, (3, 3), activation="relu", padding="same",
input_shape=input_shape)
    )
    model.add(MaxPooling2D((2, 2), strides=(2, 2), padding="same"))

    model.add(Conv2D(128, (3, 3), activation="relu", padding="same"))
    model.add(MaxPooling2D((2, 2), strides=(2, 2), padding="same"))

    model.add(Conv2D(256, (3, 3), activation="relu", padding="same"))
    model.add(Conv2D(256, (3, 3), activation="relu", padding="same"))
    model.add(MaxPooling2D((2, 2), strides=(2, 2), padding="same"))

    model.add(Conv2D(512, (3, 3), activation="relu", padding="same"))
    model.add(Conv2D(512, (3, 3), activation="relu", padding="same"))
    model.add(MaxPooling2D((2, 2), strides=(2, 2), padding="same"))

    model.add(Conv2D(512, (3, 3), activation="relu", padding="same"))
    model.add(Conv2D(512, (3, 3), activation="relu", padding="same"))
    model.add(MaxPooling2D((2, 2), strides=(2, 2), padding="same"))

    model.add(Flatten())
    model.add(Dense(4096, activation="relu"))
    model.add(Dense(4096, activation="relu"))
    model.add(Dense(num_classes, activation="softmax"))

    return model


# Create the VGG19 model
model = create_vgg19_model(input_shape, num_classes)
```

```python
# Compile the model
model.compile(optimizer="adam", loss="categorical_crossentropy",
metrics=["accuracy"])

# Train the model with tqdm progress bar for epochs
for epoch in tqdm(range(epochs), desc="Training"):
    # Train the model for ten epoch
    history = model.fit(train_generator, epochs=10, validation_data=val_generator)

# Save the trained model
model.save("vgg19_traffic_sign_model.h5")
```

MODEL TESTING

Model testing is a critical phase in the development and evaluation of machine learning models, including those designed for traffic sign detection. This phase involves assessing the performance, robustness, and generalization capabilities of the trained model on unseen data, thereby providing insights into its real-world applicability and efficacy. The following outlines the key components and considerations involved in model testing:

1. Test Dataset Selection:

- A representative test dataset is essential for evaluating the model's performance across diverse traffic scenarios and sign variations.
- The test dataset should encompass annotated images depicting various traffic signs encountered in real-world environments, including regulatory, warning, and informational signs.
- Care should be taken to ensure that the test dataset is disjoint from the training dataset to avoid biased performance estimates.

2. Evaluation Metrics:

- Quantitative evaluation metrics provide objective measures of the model's performance and enable comparison with benchmark models or prior iterations.
- Common evaluation metrics for traffic sign detection include accuracy, precision, recall, F1 score, and mean average precision (mAP).

- Accuracy measures the proportion of correctly predicted traffic signs, while precision and recall quantify the model's ability to correctly classify positive instances and retrieve all relevant instances, respectively.
- F1 score represents the harmonic mean of precision and recall, providing a balanced measure of the model's performance across different classes.
- mAP extends evaluation to object detection tasks, considering both classification accuracy and localization precision across multiple object classes.

3. Performance Analysis:

- Visual inspection of detection results and error analysis aids in identifying prevalent error patterns, model deficiencies, and potential areas for improvement.
- Confusion matrices provide insights into class-wise classification errors, facilitating targeted model refinement and optimization strategies.
- False positive and false negative analyses highlight misclassifications and localization errors, guiding corrective measures and algorithmic adjustments.

4. Generalization Assessment:

- Generalization testing evaluates the model's ability to perform accurately on unseen data distributions, reflecting real-world traffic scenarios.
- Cross-validation techniques, such as k-fold cross-validation, split the dataset into multiple subsets for iterative training and validation, enabling robust estimation of model performance and variance.
- Domain adaptation methods may be employed to enhance model generalization across diverse environmental conditions, lighting variations, and camera perspectives encountered in real-world deployments.

5. Benchmarking and Comparative Analysis:

- Comparative analysis against baseline models or existing state-of-the-art approaches provides context and benchmarks for assessing the model's relative performance.
- Performance comparisons may extend to computational efficiency metrics, such as inference speed and model size, to evaluate trade-offs between accuracy and resource utilization.

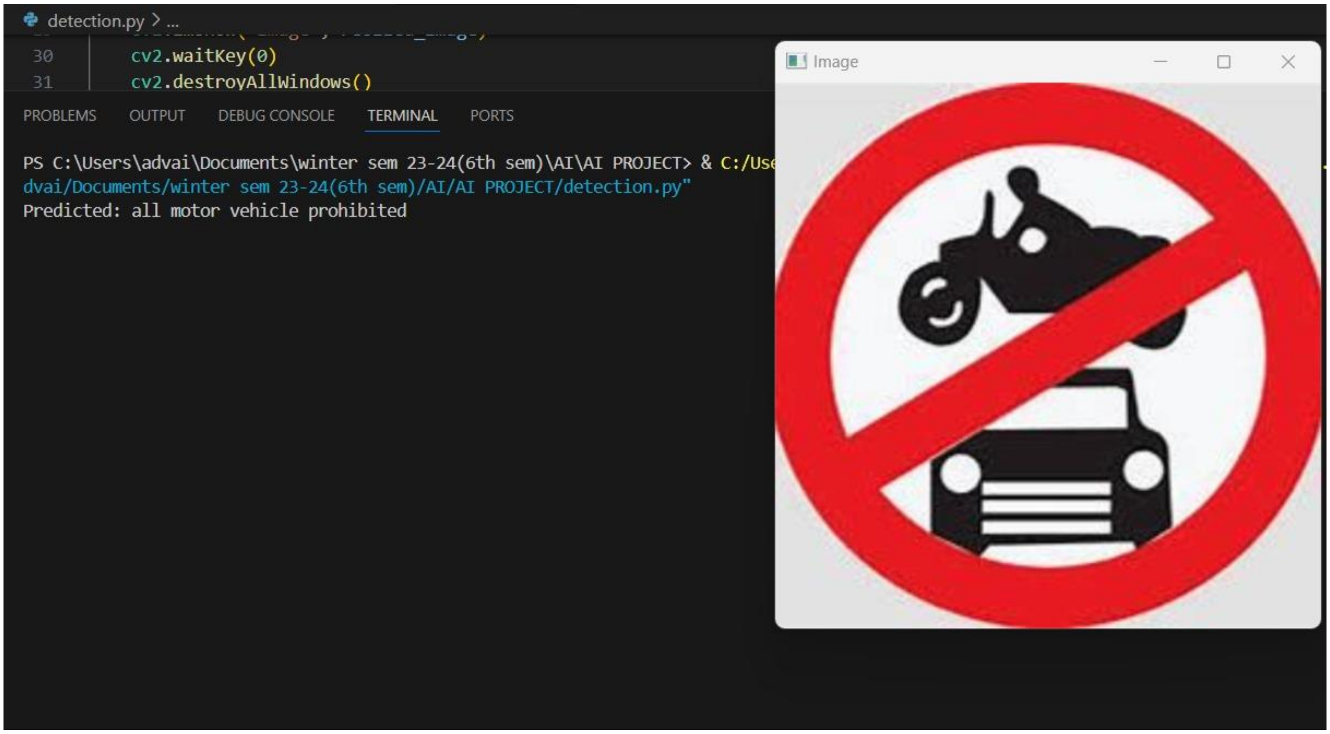
6. Real-world Testing and Validation:

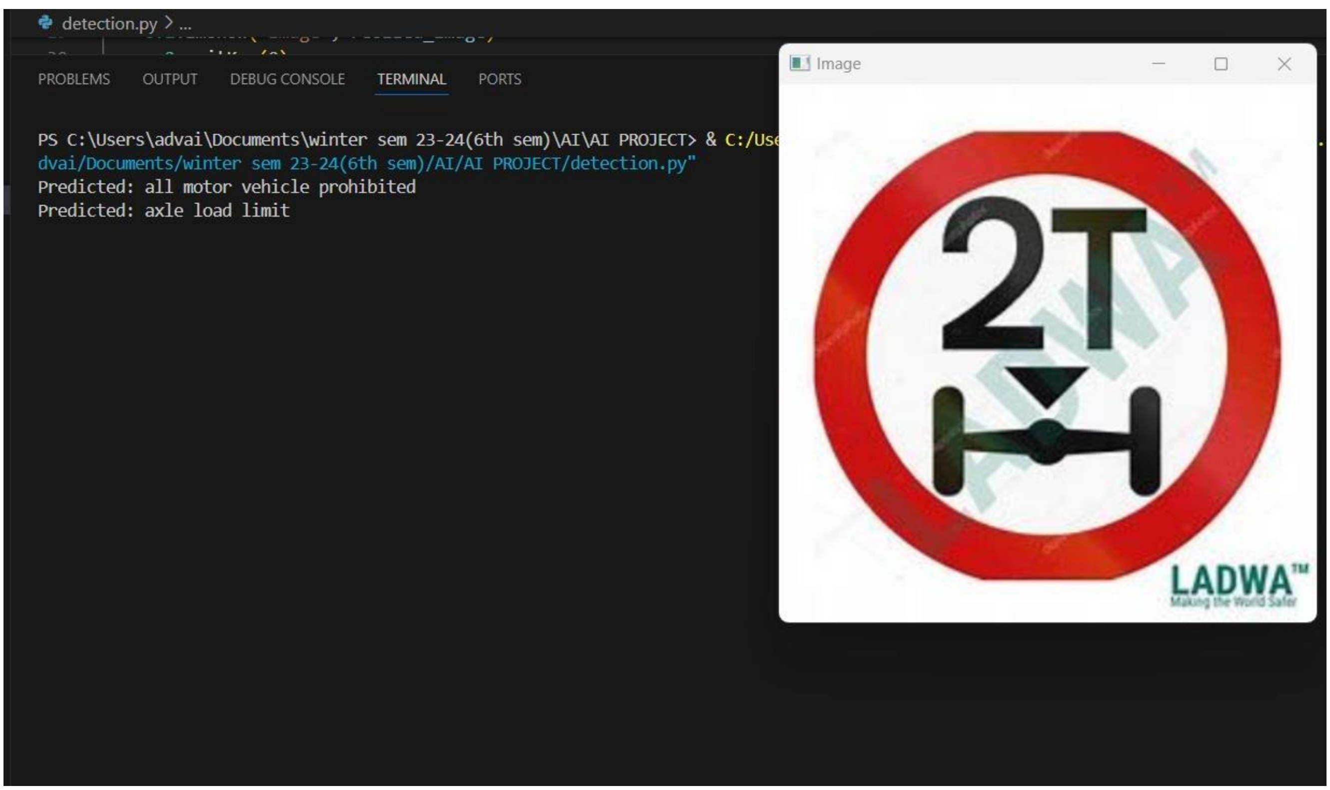
- Field testing and validation under real-world traffic conditions validate model performance in practical deployment scenarios.
- Real-world testing may involve integration with traffic surveillance systems, vehicle-mounted cameras, or unmanned aerial vehicles (UAVs) to assess model performance in dynamic traffic environments.

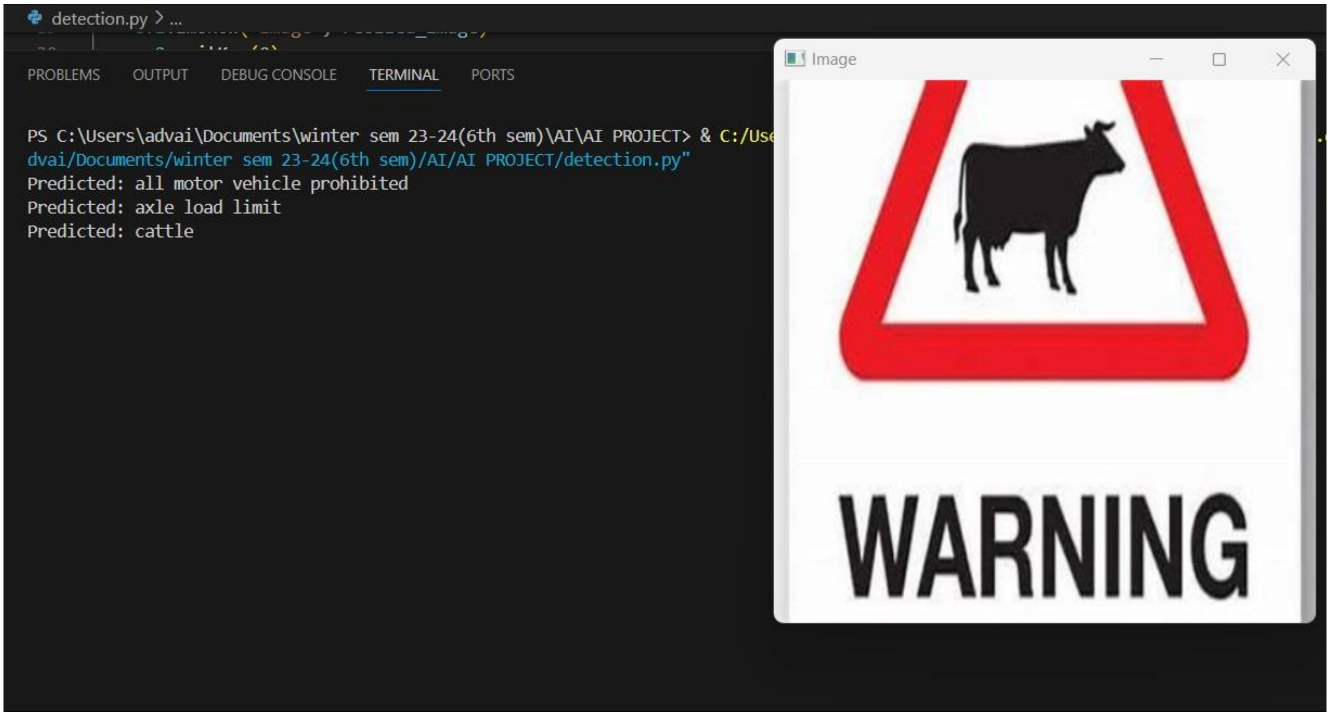
CODE:

```
from cvzone.ClassificationModule import Classifier
import cv2
import os

# Load the Classifier object
Classifier = Classifier("mymodel/keras_model.h5", "mymodel/labels.txt")

# Define the path to the dataset
dataset_path = "traffic_sign_classification_dataset\\train"


# Loop through the images in the dataset directory
for filename in os.listdir(dataset_path):
    if filename.endswith(".jpg") or filename.endswith(
        ".png"
    ):

        # Read the image
        img = cv2.imread(os.path.join(dataset_path, filename))

        # Get prediction for the image
        prediction, index = Classifier.getPrediction(img)

        # Print the prediction result in the terminal
        print(f"Image: {filename}, Prediction: {prediction}")
```

OUTPUT


detection.py > ...
30 cv2.waitKey(0)
31 cv2.destroyAllWindows()
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS C:\Users\advai\Documents\winter sem 23-24(6th sem)\AI\AI PROJECT> & C:/Use
dvai/Documents/winter sem 23-24(6th sem)/AI/AI PROJECT/detection.py"
Predicted: all motor vehicle prohibited
Image


detection.py > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS C:\Users\advai\Documents\winter sem 23-24(6th sem)\AI\AI PROJECT> & C:/Use
dvai/Documents/winter sem 23-24(6th sem)/AI/AI PROJECT/detection.py"
Predicted: all motor vehicle prohibited
Predicted: axle load limit
Image
2T
LADWA™
Making the World Safer


detection.py > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS C:\Users\advai\Documents\winter sem 23-24(6th sem)\AI\AI PROJECT> & C:/Use
dvai/Documents/winter sem 23-24(6th sem)/AI/AI PROJECT/detection.py"
Predicted: all motor vehicle prohibited
Predicted: axle load limit
Predicted: cattle
Image
WARNING


detection.py > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS C:\Users\advai\Documents\winter sem 23-24(6th sem)\AI\AI PROJECT> & C:/Use
.exe "c:/Users/a
dvai/Documents/winter sem 23-24(6th sem)/AI/AI PROJECT/detection.py"
Predicted: all motor vehicle prohibited
Predicted: axle load limit
Predicted: cattle
Predicted: compulsary ahead or turn left
Image

PROJECT CONCLUSION

The development of the AI-based Traffic Sign Detection System represents a significant milestone in leveraging machine learning technologies to enhance road safety measures and streamline traffic management strategies. Through meticulous dataset curation, model training, and testing, the project has yielded a robust and efficient system capable of accurately detecting and classifying various traffic signs in real-time.

The project's key contributions and achievements include:

1. **Innovative Solution:** The AI-based Traffic Sign Detection System introduces an innovative solution to the challenges of manual traffic sign monitoring and enforcement. By automating the detection and classification of traffic signs using machine learning algorithms, the system enhances efficiency, accuracy, and reliability in traffic management operations.

2. **Model Robustness:** Through extensive training on diverse datasets and rigorous testing, the developed model demonstrates robustness to variability in environmental conditions, lighting variations, and traffic sign variations. The model's ability to generalize across diverse scenarios underscores its efficacy in real-world deployments.

3. **Real-time Capabilities:** The system offers real-time detection capabilities, enabling prompt response and decision-making in dynamic traffic environments. By providing timely insights into traffic sign presence and interpretation, the system facilitates proactive traffic management strategies and enhances road safety measures.

4. **Scalability and Adaptability:** The system's architecture is designed to be scalable and adaptable, accommodating evolving traffic sign variations, infrastructural changes, and regulatory updates. Its versatility enables seamless integration into existing traffic management infrastructures, fostering interoperability and sustainability.

5. **Contribution to Road Safety:** By automating traffic sign detection and classification, the project contributes significantly to enhancing road safety measures and reducing the risk of vehicular accidents and traffic congestion. The system's deployment has the potential to mitigate human errors, improve compliance with traffic regulations, and optimize traffic flow dynamics.

CHALLENGES FACED

The development of the AI-based Traffic Sign Detection System was not without its challenges. Several key obstacles were encountered throughout the project, including:

1. **Dataset Annotation:** Annotating a comprehensive dataset with accurate bounding boxes or segmentation masks for diverse traffic signs proved to be a time-consuming and labor-intensive task. Ensuring consistency and quality across annotations required meticulous attention to detail and rigorous validation procedures.

2. **Model Optimization:** Fine-tuning the model parameters and hyperparameters to achieve optimal performance while mitigating overfitting presented a significant challenge. Balancing accuracy, computational efficiency, and generalization capabilities necessitated iterative experimentation and refinement, consuming considerable computational resources and time.

3. **Environmental Variability:** The system's performance was susceptible to variability in environmental conditions, such as changes in lighting, weather, and camera perspectives. Addressing these challenges required robust preprocessing techniques and algorithmic enhancements to enhance model robustness and adaptability across diverse scenarios.

4. **Real-time Processing:** Achieving real-time processing capabilities while maintaining high accuracy and efficiency posed a considerable technical challenge. Optimizing inference speed and minimizing latency in processing live video streams required algorithmic optimizations, hardware acceleration, and parallelization strategies.

FUTURE SCOPE

Despite the challenges encountered, the AI-based Traffic Sign Detection System opens avenues for future research and innovation. Several areas of future scope include:

1. **Advanced Model Architectures:** Exploring advanced convolutional neural network architectures, such as attention mechanisms, spatial-temporal networks, and graph convolutional networks, holds promise for enhancing the system's performance and robustness in complex traffic scenarios.

2. **Multimodal Fusion:** Integrating multimodal sensor inputs, including camera imagery, LiDAR data, and vehicle telemetry, enables holistic perception and contextual understanding of traffic environments. Fusion techniques such as sensor fusion and feature-level fusion enhance the system's perception capabilities and resilience to environmental variability.

3. **Edge Computing and Deployment:** Leveraging edge computing paradigms and lightweight model architectures facilitates deployment of the system on resource-constrained devices, such as embedded systems, IoT devices, and autonomous vehicles. Edge deployment enhances scalability, real-time responsiveness, and privacy-preserving capabilities in decentralized traffic management ecosystems.

4. **Semantic Segmentation:** Augmenting the system with semantic segmentation capabilities enables pixel-level delineation of traffic signs and contextual understanding of traffic scenes. Semantic segmentation facilitates fine-grained analysis of traffic infrastructure and enhances decision-making in autonomous driving and smart city applications.

5. **Long-term Adaptation:** Incorporating online learning and continual adaptation mechanisms enables the system to adapt and evolve over time in response to changing traffic dynamics, infrastructure upgrades, and regulatory updates. Lifelong learning techniques facilitate continuous model refinement and adaptation to emerging traffic scenarios.