Stop Signs Detection Using ML Model

Mahadi Melon Soykat
Department of Electrical and Computer Engineering
North South University
Dhaka,Bangladesh
mahadi.soykat@northsouth.edu

Shaznin Jahan Sudha
Department of Electrical and Computer Engineering
North South University
Dhaka,Bangladesh
shaznin.sudha@north.edu

Md Tahsinul Islam
Department of Electrical and Computer Engineering
North South University
Dhaka,Bangladesh
tahsinul.islam08@northsouth.edu

***Abstract*— A machine learning method for identifying stop signs in street photos is presented in this work with the goal of improving traffic sign identification for autonomous vehicles and traffic control systems. Annotations defining the stop sign areas inside each of the 50 street photos featuring one or more stop signs were created for the dataset. In order to make the dataset compatible with the detection model, the annotation coordinates were reformatted from the Caltech 101 dataset. A convolutional neural network (CNN) was used to train the model to automatically identify stop signs in a variety of scenarios. When the model was evaluated on a test set, it showed encouraging results, detecting stop signs with substantial accuracy and resilience. Although the model worked well in typical lighting situations, further work is required to improve identification in difficult contexts, such dim illumination. This study establishes the foundation for further improvements, such as real-time deployment in autonomous vehicle systems, and showcases the potential of machine learning in traffic sign identification.**

***Keywords*— Stop Sign Detection, Machine Learning, Traffic Sign Identification, Autonomous Vehicles, Traffic Control Systems, Image Annotation, Convolutional Neural Network (CNN), Object Detection, Real-Time Deployment, Computer Vision, Low-Light Detection, Caltech 101 Dataset.**

I. Introduction

For computer vision applications like traffic monitoring and autonomous driving, the ability to recognize things like stop signs is crucial. Creating a machine learning model to identify stop signs in street photos by anticipating their bounding box locations is the main goal of this research. To achieve accurate localization inside pictures, we use bounding box regression on a subset of the Caltech101 dataset that has stop signs tagged.

The convolutional neural network (CNN) architecture VGG16, which is well-known for image classification, serves as the foundation for our model. We modify VGG16 to become a regression model for bounding box prediction by eliminating its top layers. To make the annotations compatible with our model, they were first transformed from Matlab format (Ymin, Ymax, Xmin, Xmax) to Python format (Xmin, Ymin, Xmax, Ymax).

We put this model into practice using TensorFlow and Keras, with assistance from Scikit-learn for data preparation, OpenCV for image processing, and Matplotlib for visualizations. The model effectively detects stop signs by processing input photos and producing bounding box coordinates. This study demonstrates an object detection method, setting the stage for further advancements in computer vision applications.

II. Literature Review

The task of detecting and recognizing traffic indicators, especially stop signs, has been the subject of numerous research articles. The following important sources have impacted this project

A. Existing Research:

- *A Survey on Traffic Sign Detection and Recognition:*This survey provides a comprehensive overview of various techniques used for traffic sign detection and recognition, including traditional computer vision methods and deep learning approaches. It discusses challenges such as occlusion, varying lighting conditions, and camera noise, which are relevant to stop sign detection. [1]

- *Real-Time Traffic Sign Detection and Recognition System using Computer Vision and Machine Learning:* This paper presents a real-time system for traffic sign detection and recognition, focusing on the use of deep learning techniques. The authors demonstrate the effectiveness of CNN-based models in accurately detecting and classifying traffic signs, including stop signs. [2]

- *A Comparative Study of Deep Learning Models for Traffic Sign Detection and Recognition:* This study compares different deep learning architectures, such as Faster R-CNN, YOLOv3, and SSD, for traffic sign detection and recognition. It provides insights into the strengths and weaknesses of each model and helps in selecting the most suitable approach for the specific task of stop sign detection. [3]

B. Limitations:

Although stop sign identification has greatly improved thanks to deep learning, there are still a number of drawbacks. Accurate detection may be hampered by unfavorable weather conditions, occlusions, fluctuating illumination, poor camera quality, and computational expense. Deliberate attempts to conceal or damage indicators, along with problems with data quantity and quality, present further difficulties. Researchers are still investigating methods like data augmentation, adversarial training, and real-time adaptation to get around these restrictions.

III. Methods and Materials

A. Methodology

This stop sign detecting project's methodology consists of a number of crucial components. The Caltech101 dataset is first preprocessed to only contain images that have bounding box annotations and stop signals, formatted in Python coordinates (Xmin, Ymin, Xmax, Ymax). To accommodate the VGG16 input size, we load and resize photos using OpenCV. The VGG16 model is then set up for bounding box regression by deleting its top layers, which allows it to predict bounding box coordinates rather than picture classification. We train the model with a mean squared error (MSE) loss function that is optimized for precise coordinate prediction using TensorFlow and Keras. To monitor performance, the dataset is divided into training and validation sets, and bounding box predictions are visualized using Matplotlib. In order to increase detection accuracy, we modify the model and compare predicted bounding boxes to ground-truth annotations to assess accuracy. This pipeline illustrates the use of deep learning for object localization.

B. Materials

1) Dataset: Caltech101 Stop Sign Dataset, which was converted from Matlab format to Python coordinates (Xmin, Ymin, Xmax, Ymax) and includes pictures of streets with stop signs and annotations for bounding boxes.

2) Libraries and Software:

   a) *Python:* The project's implementation programming language.

   b) *TensorFlow and Keras:* The VGG16-based bounding box regression model is constructed and trained using the deep learning tools TensorFlow and Keras.

   c) *OpenCV :*is an image processing library that may be used to load, resize, and modify images.

   d) *Matplotlib:* A visualization tool for tracking training progress and plotting images with bounding boxes.

   e) *Scikit-learn:* For preprocessing, model evaluation, and dataset splitting.

3) VGG16 Model Architecture: Bounding box coordinates can be predicted using the VGG16 Model Architecture, a pre-trained convolutional neural network (CNN) model modified for regression.

4) Annotation File: The bounding box coordinates for every stop sign in the dataset are contained in the annotation file, which serves as the model's training ground truth.

IV. Work Process

A. *Data Collection and Preprocessing:* The Caltech101 dataset, which consists of pictures of stop signs with their annotated bounding boxes, is gathered at the start of the research. To guarantee compatibility, the annotations—which were initially in Matlab coordinates (Ymin, Ymax, Xmin, Xmax)—are changed to Python format (Xmin, Ymin, Xmax, Ymax). The photos are loaded and scaled using OpenCV to satisfy the VGG16 model's input size specifications.

B. *Model Configuration:* Configuring the VGG16 architecture is the next stage. The pre-trained model can be modified for bounding box regression instead of picture classification by loading it without its top layers. After that, custom layers are applied to produce the bounding box's four coordinates. The loss function used to optimize continuous coordinate predictions is known as the mean squared error, or MSE.

C. *Training and Validation:* In order to efficiently track the model's performance, the dataset is separated into training and validation sets. TensorFlow and Keras are used to train the model, and hyperparameters are changed for increased accuracy. Loss curve visualizations are produced during the training process, and early halting is used to avoid overfitting.

V. conclusion

To sum up, this project effectively creates a fundamental framework for stop sign detection utilizing the VGG16 model and bounding box regression. We have successfully finished the training and validation phases, set up the model architecture, and preprocessed the Caltech101 dataset. The training process has been made easier by the use of TensorFlow and Keras, producing preliminary findings that will be further assessed for correctness. Future work will concentrate on enhancing the model's performance and putting the last evaluation and testing procedures into place to guarantee reliable stop sign identification in a variety of scenarios, even though the training and validation phases are finished.

REFERENCE

[1] Nursabillilah Mohd Ali,Safirin Mohd Karis,Amar Faiz Zainal Abidin, "Traffic sign detection and recognition: Review and analysis," researchgate, December 2015. [Online]. Available: https://www.researchgate.net/publication/286762414_Traffic_sign_detection_and_recognition_Review_and_analysis. [Accessed 01 October 2024].

[2] R. Patil, "Real-Time Traffic Sign Detection and Recognition System using Computer Vision and Machine Learning," researchgate, April 2024. [Online]. Available: https://www.researchgate.net/publication/379943624_Real-Time_Traffic_Sign_Detection_and_Recognition_System_using_Computer_Vision_and_Machine_Learning. [Accessed 01 October 2024].

[3] Yanzhao Zhu,Weiqi Yan, "Traffic sign recognition based on deep learning," researchgate, May 2022. [Online]. Available: https://www.researchgate.net/publication/359079542_Traffic_sign_recognition_based_on_deep_learning. [Accessed 01 October 2024].