**An approach for traffic sign recognition**

Nguyen Tien Dat1, Vu Minh Quan1, Nguyen Hoang Thanh1, Duong Huu Thanh1\*

1Ho Chi Minh City Open University, Ho Chi Minh City, Vietnam

\*Corresponding author: thanh.dh@ou.edu.vn

|  |  |
| --- | --- |
| **ARTICLE INFO** | **ABSTRACT** |
| **DOI:**10.46223/HCMCOUJS. tech.en.  Received: August 20th, 2023  Revised: February 20th, 2024  Accepted: February 26th, 2024  *Keywords*:  YOLO: You Only Look One; Object Detection; NMS: non-maximum suppression, mAP: mean average precision, FPS: frame per second | In this article, we present a model for detecting and recognizing traffic signs based on the YOLO algorithm. Our system is capable of detecting various types of traffic signs, including prohibitory, stop, no entry, speed limit, regulatory, and hazardous signs, in real-world scenarios. However, there are still some cases where successful recognition is not achieved. Experiments were conducted on a dataset of 29,632 images, yielding a recognition accuracy of 86.8%. With relatively high accuracy, the system performs well in practical environments, yet some errors still persist during detection. |

# 1. Introduction

The application of information technology in daily life is gradually becoming more common, with people tending to adopt technology and smart city models to address social issues, and traffic issues are among the hot topics. The application of traffic sign recognition helps to address problems related to misinterpretation of traffic signs, leading to traffic congestion and preventable accidents. According to Lao Động newspaper in 2022, there were nearly 3354 road traffic accidents, with nearly 3% attributed to errors in interpreting traffic signs. Detecting and recognizing traffic signs is an intelligent tool to support safer participation in traffic. Such intelligent systems are being strongly developed worldwide, with works like [4] Jing Yu, Xiaojun Ye, and Qiang Tu's (2022) "Traffic Sign Detection and Recognition in Multi Images Using a Fusion Model With YOLO and VGG Network";... In Vietnam, there are also some research works such as [1] T.Q.Bảo, T.H.Chen, Q.Định's, "Detection and Recognition of Road Traffic Signs Using HOG Features and Artificial Neural Networks";...

This article introduces a technique for detecting and recognizing traffic signs utilizing the YOLO model. YOLO leverages state-of-the-art developments in deep learning and computer vision, delivering unmatched performance in both speed and accuracy. Its efficient architecture renders it applicable across a range of scenarios and readily deployable on diverse hardware platforms, spanning from edge devices to cloud APIs. To develop an application for traffic sign detection and recognition that meets everyday life needs, we collected and trained the model on a dataset consisting of 29,632 images, achieving an overall accuracy of 86.8%.

Throughout our research process, we have recognized that there are still numerous challenges in attempting to improve accuracy and apply the project to real-life scenarios. Firstly, to enhance accuracy, we need to search for and gather images to create a dataset with high reliability. Additionally, we must consider related conditions such as blurriness, occlusion, and weather conditions affecting the images. Once a sufficient number of scenes have been collected, we proceed to label each object, also known as preprocessing.Images containing traffic signs captured from real-world scenarios often lack proper annotations, making it challenging to tag them with the names of specific signs they contain. This limitation hinders the use of traditional methods relying on pre-tagged images. Additionally, to effectively simulate real-world scenarios, benchmark datasets must include images without traffic signs. This enables the evaluation of a detector's ability to differentiate between genuine traffic signs and similar-looking objects. After curating a diverse dataset, we train the model and assess its reliability. Through multiple training iterations, we compare and enhance dataset reliability, identifying and rectifying deficiencies to achieve optimal results. Ultimately, we select the dataset with the highest accuracy for utilization.

# 2. Related studies

In recent years, the field of traffic sign detection and recognition has witnessed significant advancements, reflecting the increasing demand for intelligent transportation systems and road safety measures. Notable studies have explored and proposed methods and techniques to enhance the accuracy and effectiveness of traffic sign recognition systems.

In India, there is a YOLOv4 research group in traffic sign recognition and detection [10]. This work introduces a modified YOLOv4-based deep learning model for robust traffic sign detection and recognition in challenging environments. By incorporating CSPDarknet53 as the backbone and employing innovative techniques such as anchor box calculation using GIoU and nighttime image enhancement, the model demonstrates superior performance. Experimental results on diverse datasets including MTSD, TT-100K, and an Indian traffic sign dataset showcase remarkable accuracy improvements over existing methods. However, there remain challenges in extending the model's performance to different datasets, as evidenced by variations in accuracy on cross-data experiments.

A notable study conducted by authors [1] proposed a method for detecting and recognizing traffic signs using Histogram of Oriented Gradients (HOG) features and artificial neural networks. This work highlights the effectiveness of combining feature extraction techniques with machine learning algorithms for accurate traffic sign recognition.

Another study [6] focused on the detection and classification of traffic signs in real-world environments. This approach contributes to addressing challenges in detecting and classifying traffic signs in real-world settings, thereby promoting the development of autonomous driving systems and intelligent transportation.

Another approach proposed is a model that combines YOLO and VGG networks for multi-image traffic sign detection and recognition [4]. The goal of this approach is to enhance the robustness and accuracy of the traffic sign recognition system in diverse real-world scenarios.

These studies all contribute positively to improving the performance and accuracy of traffic sign recognition systems, thereby enhancing traffic safety and driving forward the development of related technologies.

# 3. The research method

## 3.1 Dataset

We have collected common traffic signs in Vietnam and categorized them into 7 classes, as shown in Table 1. The total number of images in this dataset, which includes 7 classes, is approximately 29,632 images, evenly distributed across each class. We have labeled each class for every traffic sign appearing in the images. To prepare the dataset for model training, we preprocessed the data to generate additional special cases and adjusted the image sizes to enhance the model's ability to recognize the traffic signs during training.

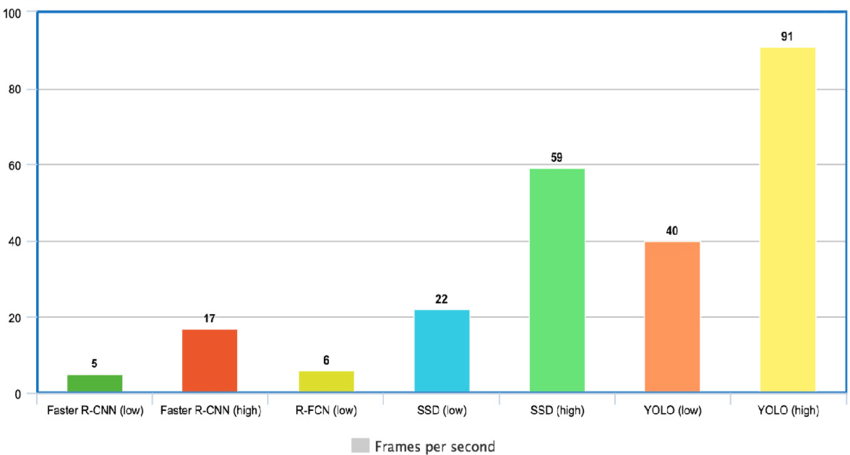
**Table 1**: A dataset for traffic sign recognition

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Example** | **Training dataset** | **Test dataset** | **Validation dataset** | **Total** | **Class** | **Label** |
| A no truck sign  Description automatically generated | 634 | 676 | 621 | 1,931 | 0 | Prohibition Traffic Sign |
| A blue and red circle with a red cross  Description automatically generated | 885 | 737 | 747 | 2,369 | 1 | No parking, stopping traffic sign |
| A red circle with a white rectangle in it  Description automatically generated | 341 | 695 | 633 | 1,669 | 2 | No entry traffic sign |
| A red circle with a black arrow pointing to the right  Description automatically generated | 792 | 349 | 358 | 1,499 | 3 | No turn traffic sign |
| A red and white sign with black numbers  Description automatically generated | 612 | 496 | 434 | 1,542 | 4 | Limit speed traffic sign |
| A blue circle with white arrows  Description automatically generated | 464 | 532 | 522 | 1,518 | 5 | Mandatory traffic sign |
| A yellow triangle sign with a person and a shovel  Description automatically generated | 547 | 1402 | 1469 | 3,418 | 6 | Danger traffic sign |

## 3.2 YOLO

For the field of object detection, there are numerous methods available, such as using HOG and artificial neural networks [1], employing Deep Learning, CNN [7]. Here, we utilize the YOLO algorithm. We choose YOLO for tackling the task of detecting and recognizing traffic signs for the following reasons:

* **Speed:**Look at the figure 1 we can see exactly the difference between YOLO and others. YOLO achieves exceptional speed due to its streamlined approach, eschewing complex pipelines. It can process images at a rapid rate of 45 frames per second (FPS). Moreover, YOLO outperforms other real-time systems by more than doubling the mean Average Precision (mAP), making it highly suitable for real-time processing tasks. Observing the image below, it's evident that YOLO's performance far surpasses that of other object detectors, reaching an impressive 91 FPS.



**Figure 1**. YOLO Speed compared to other state-of-the-art object detectors

Cre: [Comparison of frames processed per second (FPS) implementing the Faster... | Download Scientific Diagram (researchgate.net)](https://www.researchgate.net/figure/Comparison-of-frames-processed-per-second-FPS-implementing-the-Faster-R-CNN-R-FCN-SSD_fig6_342570032)

* **Detection accuracy:** YOLO significantly surpasses other state-of-the-art models in detection accuracy, exhibiting minimal background errors.
* **Good generalization:**With its latest version, YOLO has taken a step forward by offering improved generalization to new domains. This enhancement further solidifies its suitability for applications that depend on fast and robust object detection.
* **Open-source:** The decision to make YOLO open-source has spurred continuous improvements by the community. This collaborative effort has played a pivotal role in the rapid advancements of YOLO within a relatively short timeframe.

### **3.2.1 YOLOv8**

YOLO object detection system, emphasizing its adaptation for real-time surface defect detection in industrial settings. As YOLO versions progress, particularly with YOLO-v5's focus on edge deployment, research interest remains high due to its open architecture and real-time capabilities

YOLOv6 outperforms YOLOv5, YOLOX, and PPE-YOLOE in achieving higher mean Average Precision (mAP) scores on the COCO dataset across different Frames Per Second (FPS) rates. However, it's crucial to recognize that while the COCO dataset provides valuable insights, it may not perfectly reflect the performance of these models on your specific dataset.

YOLOv7 comprises several versions, with YOLOv7 being the primary model. YOLOv7-tiny is a compact variant tailored for efficient inference on edge devices. Additionally, YOLOv7-W6, commonly employed in cloud computing, is another variant available.

YOLOv8 introduces a semantic segmentation model known as YOLOv8-Seg, which utilizes a CSPDarknet53 feature extractor as its backbone, followed by a C2f module instead of the traditional YOLO neck architecture. This module is succeeded by two segmentation heads responsible for predicting semantic segmentation masks based on the input image. Similar to YOLOv8, the model incorporates five detection modules and a prediction layer in its detection heads. YOLOv8-Seg has demonstrated state-of-the-art performance across various object detection and semantic segmentation benchmarks while maintaining high speed and efficiency.

YOLOv8 can be executed via command line interface (CLI) or installed as a PIP package. Additionally, it offers multiple integrations for labeling, training, and deployment purposes.

With the improvements introduced in various versions of YOLO, it is evident that YOLOv8 stands out in object detection performance compared to the others. Therefore, our team opted to utilize YOLOv8 for traffic sign detection and recognition.

### **3.2.2 YOLO Architecture**

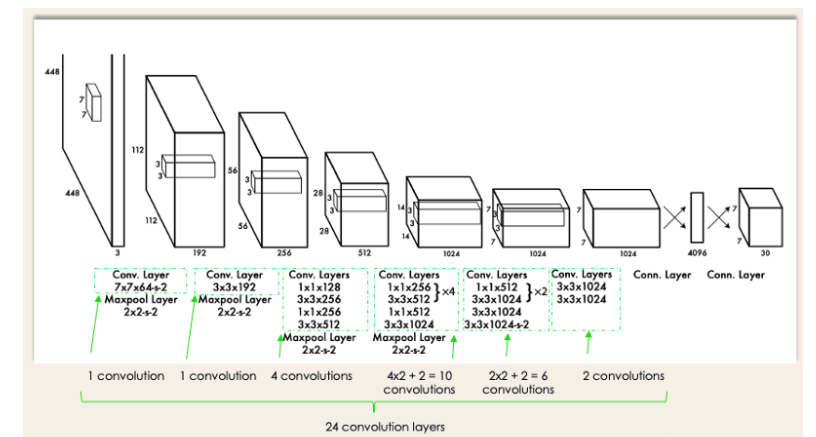
The You Only Look Once (YOLO) architecture revolutionizes object detection in images with its real-time efficiency and high accuracy. By dividing the input image into a grid and making predictions directly, YOLO eliminates the need for detecting multiple sliding windows. Its architecture comprises several crucial components working in tandem for effective object detection.

The YOLO architecture divides the input image into an S × S grid, with each grid cell responsible for detecting objects. Within each cell, the model predicts bounding boxes and associated confidence scores, as well as conditional class probabilities. This results in a tensor of predictions that encodes both object localization and class probabilities.[8]

At the heart of the YOLO architecture lie the backbone convolutional layers, forming the bedrock for feature extraction. These layers, often based on architectures like Darknet or ResNet, extract hierarchical features from the input image, capturing both low-level and high-level features crucial for accurate object detection [9].

Following feature extraction, the detection head of YOLO generates bounding boxes and associated confidence scores for detected objects through a series of convolutional layers and detection layers.

In Figure 2, the detection network architecture comprises 24 convolutional layers, succeeded by 2 fully connected layers. To condense the feature space from the preceding layers, alternating 1 × 1 convolutional layers are employed. Initially pretrained on the ImageNet classification task with an input image resolution of 224 × 224, these convolutional layers then adapt to increased resolution twofold for the detection phase.



**Figure 2: The Architecture of YOLO [8]**

The network's output manifests as a tensor of predictions with dimensions 7 × 7 × 30. This tensor encompasses the predicted bounding boxes, confidence scores, and class probabilities for the detected objects.

The bounding box prediction process entails predicting bounding boxes directly from the feature maps generated by the backbone network. Each grid cell in the feature map predicts multiple bounding boxes, accompanied by corresponding confidence scores and class probabilities, facilitating precise localization and classification of objects. Subsequently, YOLO applies non-maximum suppression (NMS) to eliminate redundant or overlapping boxes, ensuring each object is detected only once and enhancing overall precision.

During the training process, YOLO undergoes end-to-end training using labeled training data containing images annotated with bounding boxes around traffic signs. The model undergoes optimization using techniques like stochastic gradient descent (SGD) or adaptive optimization algorithms to minimize a predefined loss function, measuring the disparity between predicted and ground-truth bounding boxes. In this research, the YOLOv8 architecture was chosen for its superior performance and efficiency in traffic sign recognition tasks, building upon the advancements of previous YOLO versions by incorporating improvements in feature extraction, bounding box prediction, and model optimization.

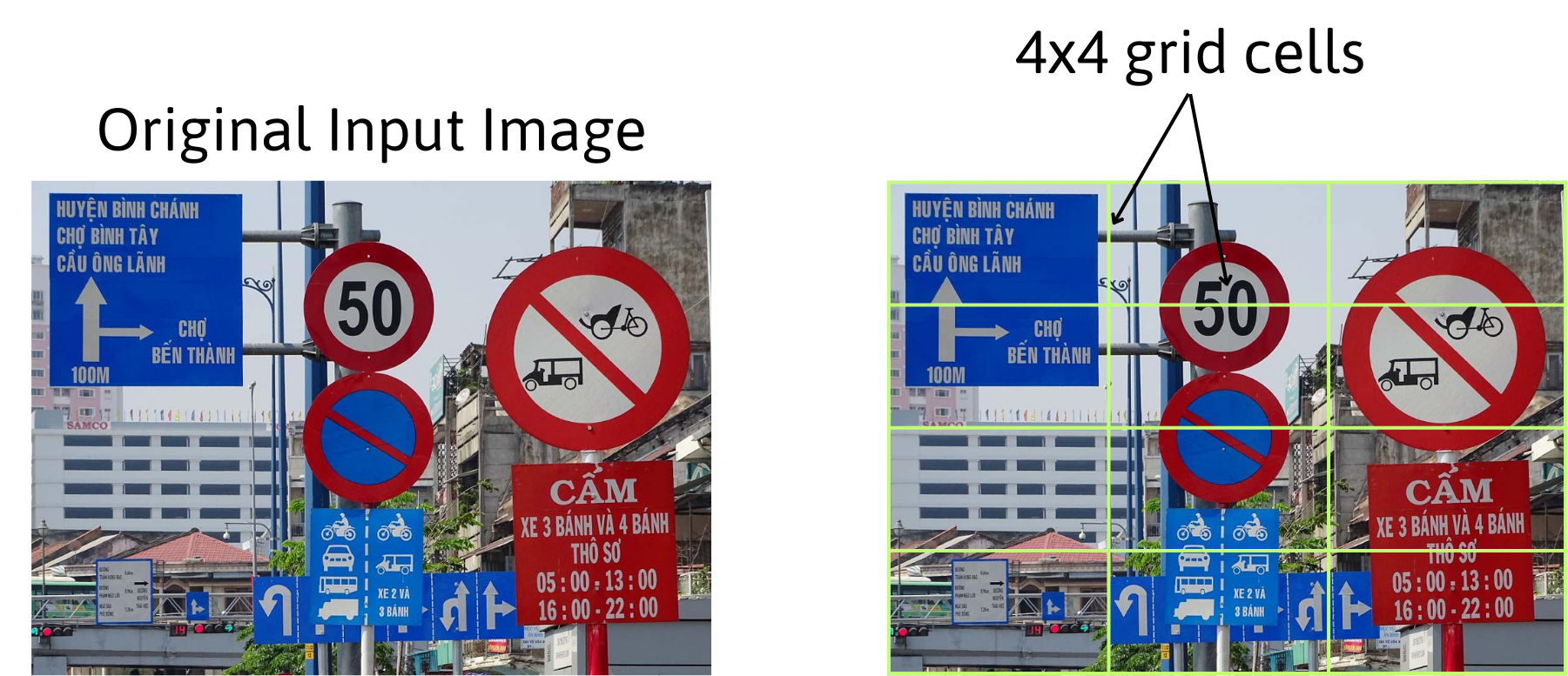
### **3.2.3 YOLO Object Detection**

In the previous section, we presented the structure of YOLO, and in this section, we will discuss how the YOLO algorithm works for detecting and recognizing traffic signs.

The YOLO algorithm works based on the following steps:

1. **Residual Blocks:**

To detect and recognize traffic signs easily, the YOLO algorithm divides the input image into an NxN grid with equal-sized cells, where N in our case is 4 as shown in the image on the right. Each cell in the grid represents localizing and predicting the class of the object lying within the cell along with the probability of each predicted object.



**Figure 3: YOLO dividing input image into 4x4 grid cells**

1. **Bounding Box Regression**

The subsequent stage involves identifying bounding boxes that outline all objects within the image. The number of bounding boxes can match the number of objects present in the image.

YOLO employs a single regression module to determine the attributes of these bounding boxes, represented by Y in the following format Y = [pc, bx, by, bh, bw, c1, c2]

This process is particularly crucial during the model's training phase.

The element pc denotes the probability score of the grid containing an object. For example, all grids highlighted in red will possess a probability score greater than zero (significant). The image on the right is a simplified version, as the probability of each yellow cell is zero (insignificant).

****

**Figure 4. YOLO determine the bounding box of object base on the probability**

1. **Intersection Over Unions or IOU for short**

Object detection algorithms can be divided into two main categories: single-shot detectors and two-stage detectors.YOLO is a single-shot detector that uses a fully convolutional neural network to process an image. The goal of the IOU (a value between 0 and 1) is to discard such grid boxes to only keep those that are relevant. Here is the logic behind it:

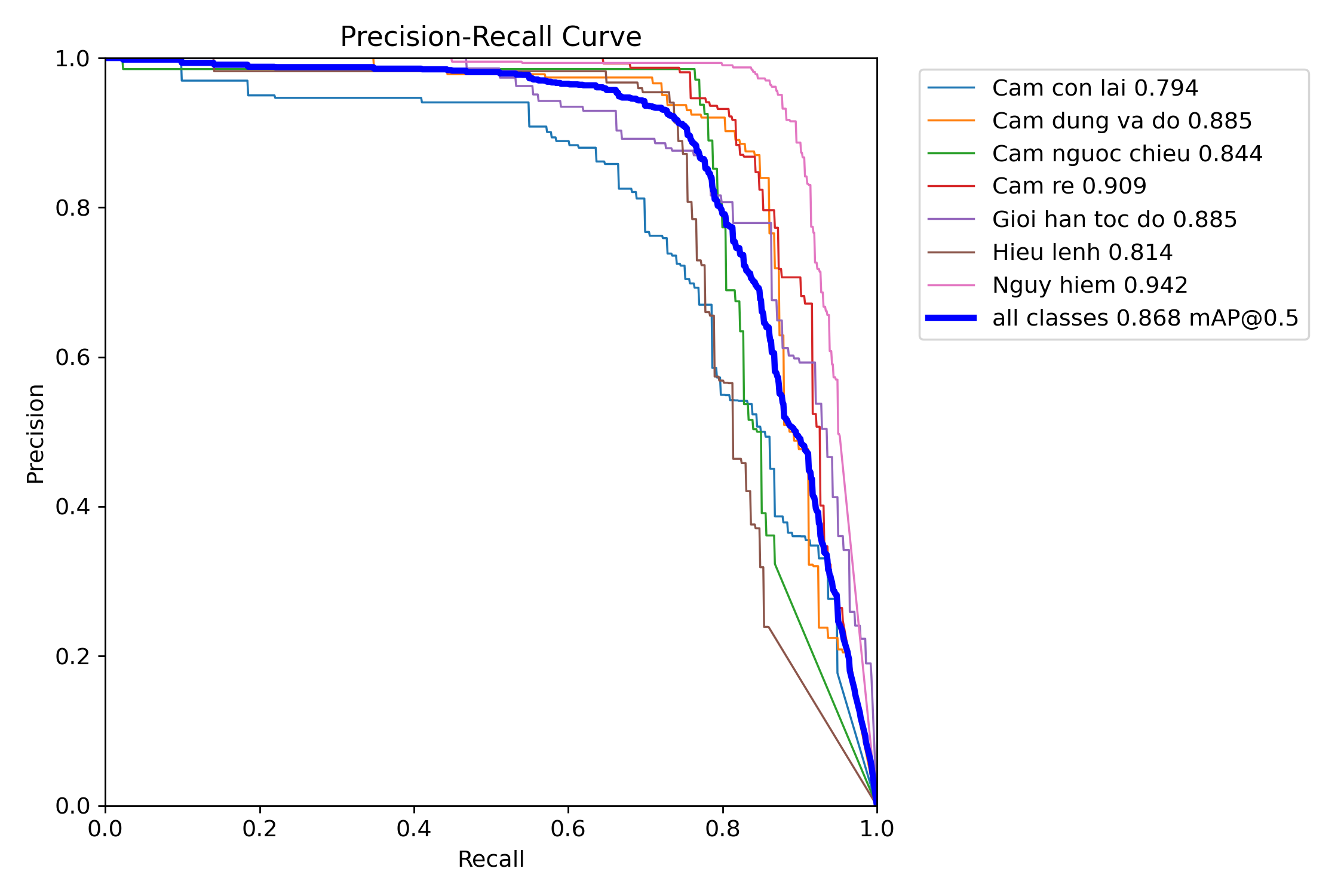
* The user defines its IOU selection threshold, which can be, for instance, 0.5.
* Then YOLO computes the IOU of each grid cell which is the Intersection area divided by the Union Area.
* Finally, it ignores the prediction of the grid cells having an IOU ≤ threshold and considers those with an IOU > threshold.

1. **Non - Maximum Suppression**

Merely establishing a threshold for the Intersection over Union (IOU) isn't always sufficient, as an object might generate multiple boxes with IOU surpassing the threshold. Retaining all such boxes could lead to noise. This is where Non-Maximum Suppression (NMS) comes into play, allowing us to retain only the boxes with the highest probability score of detection.

# 4. Results and discussions

## 4.1 Training Results



**Figure 5: Precision - Recall (mAP) Curve**

In the task of traffic sign recognition and detection, Precision-Recall metrics are crucial. Figure 5 provides an overview of the model's accuracy for each class and all classes combined. This curve aids in selecting the optimal threshold to maximize both metrics.

**Table 2. mAP of each class in traffic sign recognition**

|  |  |  |
| --- | --- | --- |
| **Class Name** | **True Prediction** | **False Prediction** |
| Another traffic sign | 79.4% | 20.6% |
| No parking, stopping traffic sign | 88.5% | 11.5% |
| No entry traffic sign | 84.4% | 15.6% |
| No turn traffic sign | 90.9% | 9.1% |
| Limit speed traffic sign | 88.5% | 11.5% |
| Mandatory traffic sign | 81.4% | 18.6% |
| Danger traffic sign | 94.2% | 5.8% |

To be applicable in practical situations, it is perhaps necessary to achieve a higher level of accuracy, including data pertaining to adverse weather conditions and environmental factors. However, overall, this is still quite an impressive outcome for initial steps.

## 4.2 Experiments

With an accuracy rate of up to 86.8% in detecting and recognizing traffic signs, we have conducted some experimental runs by predicting signs on images and videos to evaluate the model more objectively. Here are some trial results:

# 

**Figure 6: Some experimental images of traffic sign recognition**

Looking at the trial results above, we can see that the model's accuracy in recognizing and detecting traffic signs is very high, at 80% or above. However, there are still cases where images are too small or too blurry, leading to incorrect identification and lower accuracy. With such trial results in mind, we can consider applying the traffic sign detection and recognition model in real-world scenarios in the future, but adjustments will still be needed to better adapt it to the traffic conditions in Vietnam.

# 5. Conclusions and Developments

## 5.1 Conclusions

In our research, we applied the YOLO algorithm for automatic detection and recognition of traffic signs. The adoption of YOLO has made the research project more effective in modeling traffic sign detection and recognition, while improving processing speed and accuracy compared to manual methods used in previous studies.

During the research process, the YOLOv8 version has demonstrated its performance by significantly improving upon previous versions of YOLO. Utilizing YOLOv8, our team conducted the process of traffic sign recognition more quickly and accurately, contributing to enhancing traffic safety and promoting the development of automation applications in the transportation field.

We believe that this research will contribute to improving traffic safety and advancing the development of automation applications in transportation, thereby enhancing overall road safety and accelerating the growth of automation applications in the transportation sector.

## 5.2 Developments

A potential direction for the development of this research is to focus on optimizing the performance of the YOLOv8 model. This includes tuning the model's hyperparameters, expanding training data, and experimenting with new techniques to improve the accuracy and processing speed of the model.

In addition to traffic sign recognition, the model should also be applied to detect and classify other objects on the road, such as pedestrians, vehicles, or even street scenery. This could significantly enhance street surveillance and security systems.

Furthermore, research should integrate the YOLOv8 model with other methods such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), or Reinforcement Learning (RL) techniques to improve the performance and accuracy of the system.

Finally, to ensure the stability and effectiveness of the model in real-world scenarios, deployment and testing on actual systems such as autonomous vehicles and traffic surveillance cameras are necessary.

From optimizing the model, expanding its application to enhancing training data and integrating with other methods, we believe that continuing to develop and improve the YOLOv8 model will positively contribute to improving traffic safety and advancing automation applications in this field.

**References**

|  |  |
| --- | --- |
| [1] | Bảo, T. Q., Chen, T. H., & Định, T. Q. “PHÁT HIỆN VÀ NHẬN DẠNG BIỂN BÁO GIAO THÔNG ĐƯỜNG BỘ SỬ DỤNG ĐẶC TRƯNG HOG VÀ MẠNG NƠRON NHÂN TẠO”. *Tạp chí Khoa học Đại học cần Thơ*, (CĐ Công nghệ TT), 47-54, 2015. |
| [2] | Tianjiao Liang, Hong Bao, Weiguo Pan, Feng Pan, "Traffic Sign Detection via Improved Sparse R-CNN for Autonomous Vehicles". *Journal of Advanced Transportation*, vol. 2022, Article ID 3825532, 16 pages, 2022. |
| [3] | M. Flores-Calero et al., "Traffic Sign Detection and Recognition Using YOLO Object Detection Algorithm: A Systematic Review," *Mathematics*, vol. 12, no. 2, p. 297, 2024. |
| [4] | Jing Yu, Xiaojun Ye, Qiang Tu, "Traffic Sign Detection and Recognition in Multi Images Using a Fusion Model With YOLO and VGG Network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 16632-16642, 2022. |
| [5] | A. Sugiharto and A. Harjoko, "Traffic sign detection based on HOG and PHOG using binary SVM and k-NN,"in *2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, pp. 1-6, 2016. |
| [6] | Zhe Zhu, Dun Liang, Songhai Zhang, Xiaolei Huang, Baoli Li, and Shimin Hu, "Traffic-Sign Detection and Classification in the Wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2110-2118, 2016. |
| [7] | A. B. Amjoud and M. Amrouch, "Object Detection Using Deep Learning, CNNs and Vision Transformers: A Review," *IEEE Access*, vol. 11, pp. 35479-35516, April 10, 2023. |
| [8] | J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779-788, 2016. |
| [9] | J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680-1716, 2023. |
| [10] | S. Saxena, S. Dey, M. Shah, and S. Gupta, "Traffic sign detection in unconstrained environment using improved YOLOv4," Expert Systems with Applications, vol. 238, pt. A, pp. 121836, Mar. 15, 2024. |



Creative Commons Attribution-NonCommercial 4.0 International License.