Performance Comparison in Traffic Sign Recognition Using Deep Learning

Binh Dang Hai[1], Huu Duc Nguyen[1], Thanh Nam Vo[1], Phuong-Nam Tran[1], Cuong Tuan Nguyen[2], and Duc Ngoc Minh Dang[1(✉)]

[1] Computing Fundamental Department, FPT University, Ho Chi Minh City, Vietnam
{binhdhse161063,duchnse161439,namvtse161434,namtpse150004}@fpt.edu.vn, ducdnm2@fe.edu.vn

[2] Faculty of Engineering, Vietnamese-German University, Binh Duong, Vietnam
cuong.nt2@vgu.edu.vn

Abstract. In recent years, along with the increase in private cars, traffic signs have increased in quantity, demanding greater attentiveness from drivers. Many studies have been conducted on Traffic Sign Recognition (TSR) to enhance road safety and driver assistance systems by enabling vehicles to autonomously detect and interpret traffic signs, providing crucial information to drivers in real-time. This paper examines various traffic sign detection and classification models, which give practitioners and researchers valuable insights into selecting optimal solutions tailored to the needs of real-world applications. Notably, the investigation highlights YOLOv8 as a leading detection model, displaying exceptional results with an mAP of 99.4%. YOLOv8 provides various model sizes allowing for adaptation to specific real-time processing scenarios. On the other hand, the LeNet model is a standout performer in the classification domain, consistently achieving a remarkable accuracy of 98.2% while using only 0.4 million parameters. The LeNet architecture ensures accurate and rapid traffic sign classification, making it an appealing choice for applications where resource efficiency is critical.

Keywords: Faster R-CNN · TSR · CNN · YOLO · ResNet50 · VGG16 · VGG19 · LeNet

1 Introduction

Cars have become an integral mode of mobility for people as the human population has grown and our societies have developed. In recent years, there has been a significant increase in car sales, indicating the growing reliance on automobiles. However, the rise in car usage has inevitably given rise to various challenges in road traffic management. Among these challenges, one crucial aspect is the need for effective TSR systems to enhance safety and efficiency on the roads. Even though there are traffic signs posted all over the route to guide us through the

N.-S. Vo et al. (Eds.): INISCOM 2024, LNICST 595, pp. 122–138, 2024.
https://doi.org/10.1007/978-3-031-67357-3_9

real scenario as shown in Fig. 1, the growing number of vehicles complicates our traffic and makes us more perplexed. The proliferation of vehicles also causes various societal issues, including the pollution caused by vehicle exhaust, which lowers air quality and endangers the ecosystem. Due to poor visibility caused by extreme weather, particularly haze, and fog, it is simple for cars to miss crucial traffic warnings. Because of this, drivers will be more aware of breaking traffic laws, which might lead to accidents.

Fig. 1. Normal weather

Recently, there has been a rise in the number of traffic accidents. The World Health Organization found that over 1.19 million people died from these accidents in 2023. Additionally, around 20 to 50 million people were injured, and many of them ended up with disabilities because of how serious their injuries were. To combat the global surge in accidents, professionals and academics are investigating this issue as mentioned in [1]. Given that road transportation remains the primary mode for nations with vast geographic areas, research on traffic management systems like the Intelligent Traffic System is becoming crucial for the future of road traffic. The foundation of TSR [2] lies in pattern recognition, image processing, computer vision, and related technologies. Its goal is to automatically collect real-time road data while a vehicle is in motion. The correct interpretation of effective traffic sign information is essential for safe driving and maintaining smooth roads. TSR [3,4] has garnered significant attention due to its rapid expansion. However, ecological deterioration, leading to

adverse weather conditions like haze, poses a challenge to intelligent transportation systems. TSR algorithms struggle in such conditions, primarily due to poor image quality and the loss of crucial image data [5]. Therefore, it's imperative to investigate how to recognize traffic signs in hazy environments.

While many techniques exist for identifying objects in driving assistance systems, not much attention has been given to recognizing road traffic signals. Developed countries have advanced standards for their traffic signs, which provide important information for driving systems. If these signs can be recognized efficiently, it can help reduce traffic congestion and lower the chances of accidents. Deep learning, especially in image recognition, has shown great potential and is the main focus of this research. With the constant progress in the automotive industry, it is important to investigate methods for detecting speed limit signs even in challenging conditions like foggy weather. Accurately identifying traffic signs in adverse conditions has promising practical applications, and advancements in deep learning techniques are crucial for scientific and technological progress in transportation. [6] is a recent study that concentrates on detecting and classifying traffic signs in the streets of Vietnam. The researchers achieve this by comparing different algorithms that are used for detecting objects. They assess the performance of these algorithms on their own dataset called VTSDB46. However, there is a challenge with their current architecture. It becomes difficult to update the model whenever a new traffic sign needs to be added. To solve this problem, our paper examines different methods and compares their performance on the streets of Vietnam. Additionally, we propose a combination method for future development in the field of TSR.

Our primary contributions in this article cover three main aspects:

1. We propose a combined model using YOLOv8n and LeNet, which are easy to maintain and develop for TSR tasks.
2. We introduce two datasets in the TSR field: the GoPro Viet Nam Street dataset (GPVNS) for detection tasks and the Internet Traffic Sign 29 dataset (ITS29) for classification tasks.
3. We assess and compare the currently popular models using GPVNS, ITS29, and other datasets.

The rest of this paper is structured as follows. In Sect. 2, we discuss previous research on TSR and object detection. Section 3 presents our initial dataset and gives an overview of different models used for object detection and classification. Then, in Sect. 4, we present the results of these models on the dataset and compare their performance. Finally, in Sect. 5, we conclude our study by identifying the best-performing model and discussing its potential applications in TSR.

2 Related Works

2.1 Traffic Signs Detection and Recognition

Numerous studies have been conducted on the topic of Traffic Sign Detection and Recognition, aiming to improve the performance of both detection and recognition models. In 2019, Ravindran et al. [7] presented their evaluation of TSR with

a focus on selecting Deep Neural Networks that demonstrate application-oriented performance, as assessed by mAP metrics. However, distinguishing certain traffic signs poses significant challenges due to various factors, such as constraints imposed by the training dataset, variations in shape, and complexities related to text features. To address this, the authors propose the use of a second text identification technique alongside the deep neural networks. This additional approach involves recognizing text within the Region of Interest (ROI) in an image, which leads to improved accuracy in traffic sign classification. Nevertheless, it is important to note that there are still limitations in terms of speed and real-time performance.

In addition to the challenges above, adverse weather conditions, varied lighting, and limited visibility present further obstacles to traffic sign detection and recognition. To address these challenges, [8] proposes an efficient method for real-time detection and recognition of traffic signs. Their study focuses on advanced multi-object detection systems, specifically the SSD [9] and Faster R-CNN [10]. These systems incorporate feature extractors such as MobileNet v1 [11] and Inception v2 [12] to specifically tackle the challenges associated with traffic sign detection. By leveraging these advanced techniques and models, the researchers aim to improve the accuracy and robustness of traffic sign detection and recognition, even in adverse weather conditions and low visibility scenarios.

2.2 Detection Models

You Only Look Once (YOLO) [13] is an object detection algorithm that was first introduced in 2016. It revolutionized the field of computer vision by providing a fast and accurate way to detect objects in real-time video streams or images. Before YOLO, object detection algorithms typically used a two-step approach such as Faster R-CNN [10] which involved generating region proposals or candidate bounding boxes and then classifying those proposals to identify objects. However, this approach was computationally expensive and time-consuming. YOLO took a different approach by formulating object detection as a single regression problem. Instead of generating region proposals, YOLO divided the input image into a grid and predicted bounding boxes and class probabilities directly from the grid cells. This allowed YOLO to achieve real-time object detection by making predictions in a single pass through the network. Since its introduction, researchers have developed various versions of YOLO, each with improvements and advancements to enhance the algorithm's performance and accuracy.

Object detection algorithms play a vital role in the field of traffic sign detection and recognition, and continuous research efforts aim to improve both the speed and accuracy of these algorithms. Researchers are exploring lightweight network architectures and optimized inference techniques to achieve faster real-time processing and improve detection accuracy. The emergence of various efficient detection models such as YOLOv5 [14], has further contributed to the development of fast and secure object detection algorithms, making them highly effective for real-life applications in traffic sign detection and recognition.

3 Methodology

3.1 Datasets

German Traffic Sign Detection Benchmark. The German Traffic Sign Detection Benchmark (GTSDB) holds significant importance in the realms of computer vision, pattern recognition, and image-based driver assistance. It consists of 43 classes of traffic signs labeled from 1 to 43, as depicted in Fig. 2. This benchmark was introduced during the IEEE International Joint Conference on Neural Networks in 2013 and serves as a foundational dataset for researchers in the field. The distribution and labeling of the GTSDB dataset can be seen in Fig. 3. The GTSDB dataset contains 900 images categorized into 600 images for training and 300 images for testing. It specifically addresses the challenge of single-image traffic sign detection. The dataset classifies traffic signs into three distinct types and is designed to accommodate varying sign counts, sizes ranging from 16×16 to 128×128 pixels, and challenging lighting and perspective conditions.

Fig. 2. Traffic sign labels in GTSDB and GTSRB

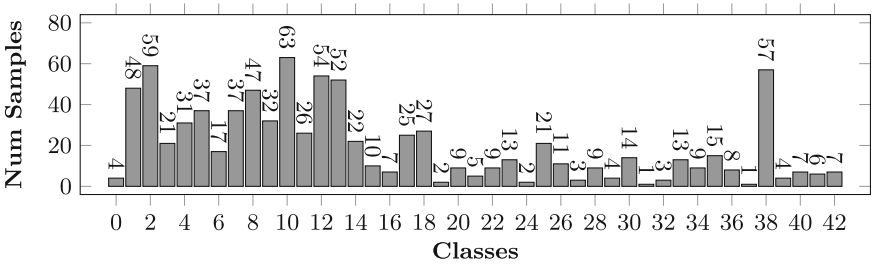

Fig. 3. GTSDB classes distribution

Another notable aspect of the GTSDB is its online evaluation system, which enables real-time analysis of detection results. This system provides a robust platform for assessing detection algorithms across diverse scenarios. The images in GTSDB are stored in PPM format, while annotations are meticulously detailed in CSV files. These annotations include crucial information such as ROI coordinates and sign class IDs, offering comprehensive insights for researchers and practitioners. The GTSDB, with its rich variety and detailed annotations, stands as a cornerstone dataset for advancing research in the field of traffic sign detection.

German Traffic Sign Recognition Benchmark. The German Traffic Sign Recognition Benchmark (GTSRB), initially introduced at the International Joint Conference on Neural Networks in 2011, stands as a prominent evaluation platform for single-image, multi-class classification challenges. Featuring over 40 distinct classes and a dataset comprising 50,000 realistic images, this benchmark welcomes participants from diverse fields without requiring specialized domain knowledge. The distributions of training and testing subsets of the GTSRB are illustrated in Figs. 4 and 5. The primary focus of this benchmark is to address the complexities associated with single-image classification, particularly in the context of traffic signs.

The dataset is meticulously designed to exhibit real-world diversity, with unique instances capturing various scenarios. The training set is organized by class, and annotations are provided in CSV files, offering detailed information for each track. For edge-based approaches, each track consists of 30 images of a single physical traffic sign, with a 10% border around the sign. The images are stored in the PPM format, featuring sizes ranging from 15×15 to 250×250 pixels. The annotation format includes essential details such as filename, dimensions, and bounding box coordinates, facilitating a comprehensive evaluation of classification algorithms on a diverse and realistic traffic sign database.

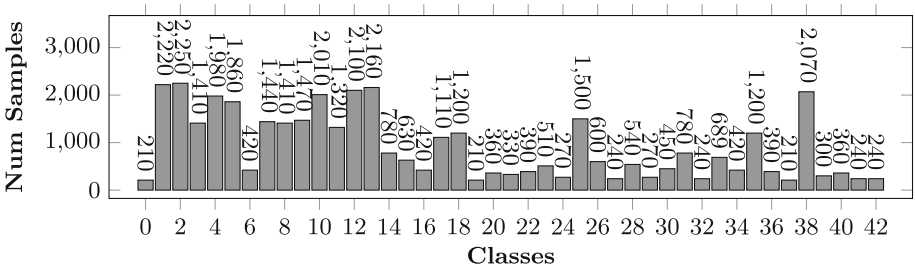

Fig. 4. GTSRB train classes distribution

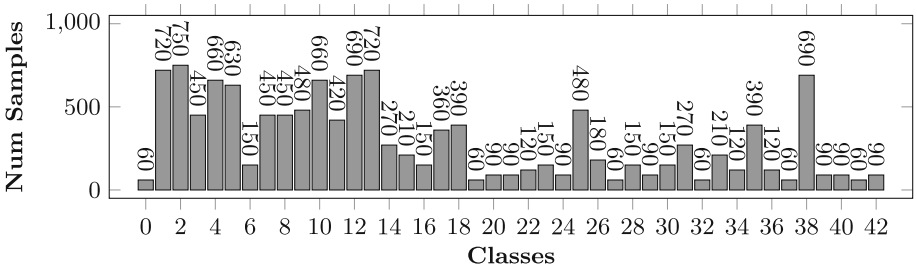

Fig. 5. GTSRB test classes distribution

GoPro Viet Nam Street Dataset. In this paper, we present the GoPro Viet Nam Street (GPVNS) dataset, which is a collection of labeled images captured from the streets of Vietnam. The data was obtained using a camera that produced frames with dimensions of 1920×1080 and 2704×1520, at a frame rate of 60 FPS. We recorded this data between 11 AM and 4 PM in various regions of Ho Chi Minh City, Viet Nam, with a focus on districts 1 and 3, known for being the most densely populated areas in the city. After segmenting the recorded video into individual frames and discarding irrelevant scenes without any traffic signs, we compiled a dataset consisting of approximately 22,000 images, totaling a size of 11.5 GB. Figure 6 shows the detail of the traffic sign and its label in the GPVNS dataset. GPVNS dataset comprises 1088 images, which contain 1615 annotations, divided into 29 classes in total. These images were carefully sorted within approximately 22000 frames recorded and were labeled by hand using the Labelbox application. Figure 7 provides an overview of the dataset's distribution, offering insights into the frequency of different classes present in the GPVNS dataset.

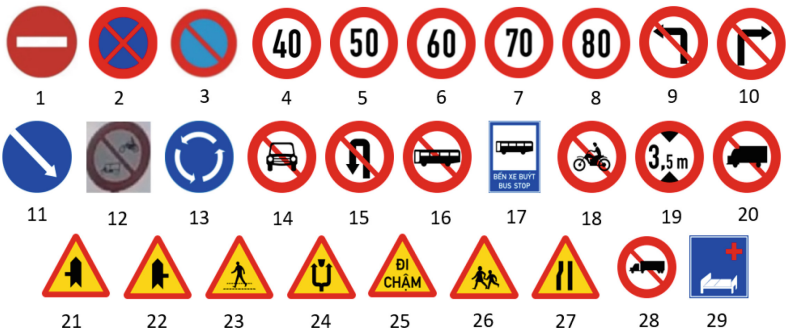

Fig. 6. Traffic sign labels in GPVNS

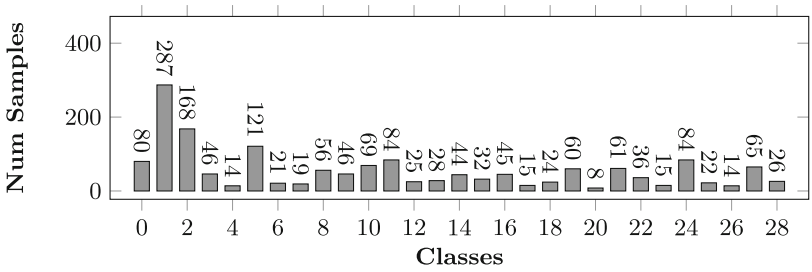

Fig. 7. GPVNS classes distribution

Internet Traffic Sign 29 Dataset. For the classification model, we present the Internet Traffic Sign (ITS29) dataset, specifically designed for classification models. The ITS29 dataset comprises approximately 28,000 images obtained from the internet, focusing on cropped traffic signs. These images are categorized into 29 distinct classes, as depicted in Fig. 6. To evaluate the performance of the classification model, we divided the dataset into training and testing subsets. Figure 8 illustrates the distribution of the training dataset, while Fig. 9 shows the distribution of the test dataset within the ITS29 dataset. These figures provide valuable insights into the composition and distribution of the training and testing data, enabling researchers to assess the generalization and performance of their classification models effectively.

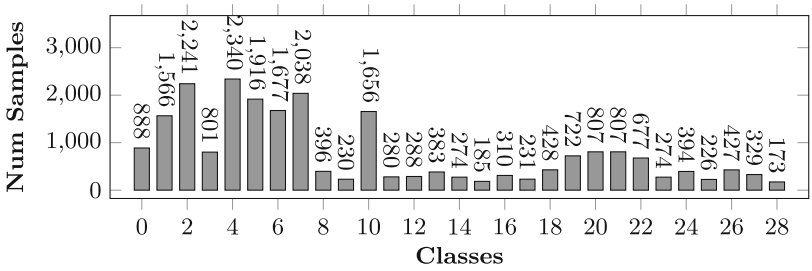

Fig. 8. ITS29 train classes distribution

3.2 Detection Model

Visual object detection is one area where deep learning has numerous practical applications. Its objective is to identify and locate objects within images, a process involving two primary steps: determining the object's position within the image, and determining its category. However, this process is complex due to the wide range of object sizes, orientations, and placements in high-resolution images. Traditionally, the approach to this problem is to create a deep neural

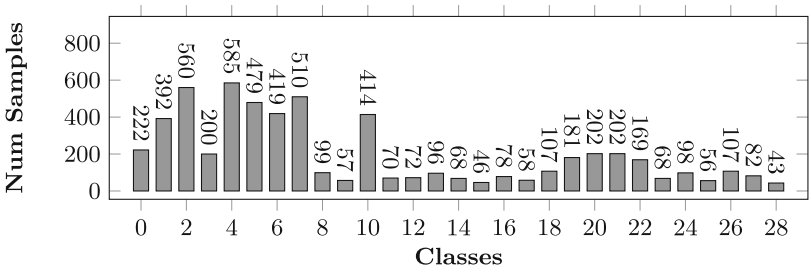

Fig. 9. ITS29 test classes distribution

network that takes image data, object labels, and positions as input. The network employs convolutional neural layers and two fully connected layers to classify objects, precisely locate them, and provide the object's exact coordinates within the image. To enhance the effectiveness of object detection, numerous deep learning models have been introduced, including R-CNN [15] and its subsequent iterations such as Fast R-CNN [16], Faster R-CNN [10], and regression-based techniques such as YOLO and SSD [9]. This section is dedicated to offering a comprehensive overview and analysis of the Faster R-CNN and YOLO models.

Faster R-CNN Model. Faster R-CNN [10] is a significant improvement over its predecessor, Fast R-CNN [16]. Instead of using the traditional selective search algorithm to extract possible object regions, Faster R-CNN uses the RPN [17] to identify candidate regions. The RPN system delivers a mix of candidate regions with varying qualities which enhances the network's recall performance, recognition rate, and precision. Figure 10 illustrates the Faster R-CNN architecture, showcasing the integration of the RPN into the overall structure.

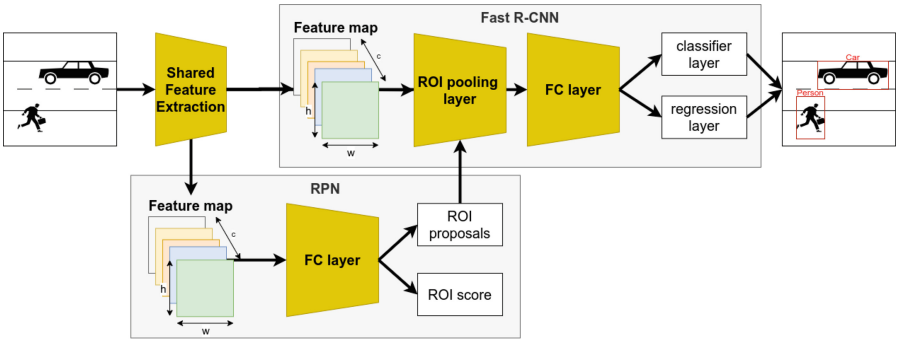

Fig. 10. Process Flow of Faster R-CNN

The specific process is described below:

1. When an image is fed into the CNN network, it results in the derivation of a convolutional layer. This convolutional layer serves a dual purpose: it acts as input for the RPN network and is also directed towards a specific, fixed convolutional layer to produce a more intricate convolutional layer feature map.
2. The RPN network delivers both scores and corresponding region recommendations. These scores are further refined using non-maximum suppression [18] with a threshold of 0.7. This process yields fewer than 300 high-quality candidate frames, which are then standardized in size through the ROI pooling layer.
3. The ROIs selected in step 2 and the high-dimensional feature map from step 1 are directed into the ROI pooling layer, which enables the extraction of region-specific features.
4. The candidate features from step 3 are fed into the fully connected layer, which computes scores and boundary regression for each candidate region. This involves a joint training process, combining soft-max loss and smooth L1 loss to handle both classification and boundary regression.

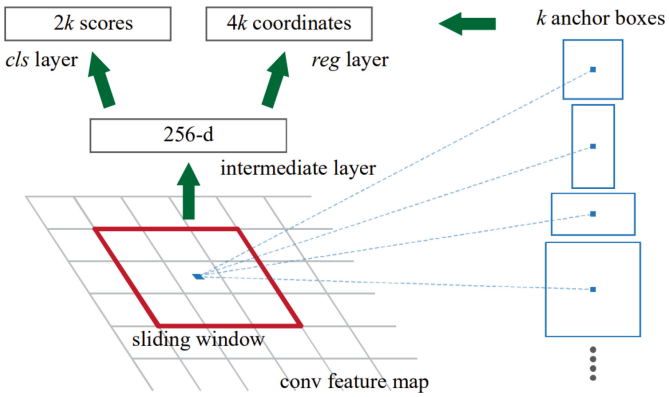

Fig. 11. Region Proposal Network

RPN uses bounding box regressions and anchors, also known as candidate boxes. It slides over the final convolutional layer to generate candidate regions at multiple scales. Figure 11 provides a visual representation of the RPN network structure. The underlying network architecture is ZF [19], and the output dimension of the conv5 layer is 256, corresponding to 256 feature maps. For each sliding window, RPN predicts two candidate regions. The classification layer outputs two scores, representing the likelihood of an object being present, and provides four sets of coordinates that correspond to the object's boundary. Due to its unique architecture, the Faster-RNN algorithm is classified as a two-stage approach to object detection.

YOLO Model

Development of YOLO Family. YOLO [13], initially introduced in 2016, is an outstanding regression algorithm for object detection in deep learning. YOLO is a representative of a single-stage algorithm as shown in Fig. 12. YOLO leverages the information from the preceding feature map to integrate classification and re-localization into a single module. This unique approach enables YOLO to achieve faster processing times compared to two-stage approaches, while still maintaining high accuracy levels. The team behind YOLO improved the algorithm over the years and released YOLOv2 [20] successively. YOLOv5 [14] was released on the GitHub platform more than a month after the release of YOLOv4, and YOLOv5.1.0 was officially online in June. These releases have contributed to the ongoing development and evolution of YOLO as a powerful tool for object detection in the field of computer vision. In June 2023, one year after the release of YOLOv5. Just a month later, the same research team introduced YOLOv7 [21], also known as MT-YOLOv6. This model established a new state-of-the-art for real-time object detection, showcasing significant advancements in the field. Finally, the most recent version, YOLOv8, was launched on January 10, 2023. This version introduces several innovations, including a new backbone network, a design that simplifies performance comparisons with previous YOLO models, a novel loss function, and an anchor-free detection head. While YOLOv8 is being regarded as the new state-of-the-art, an official paper has not been provided.

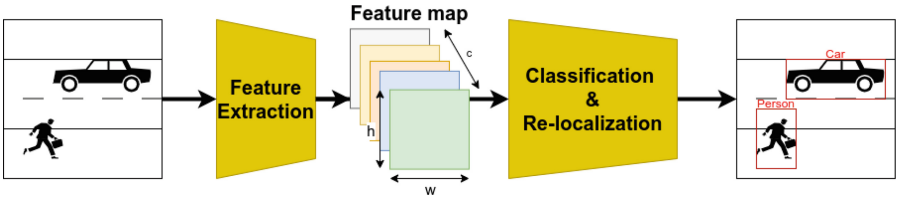

Fig. 12. Single-stage object detection algorithm

YOLOv8 for Traffic Sign Recognition. Unlike some counterparts, YOLO [13] doesn't involve a regional proposal step. In contrast to traditional CNNs, YOLO streamlines the task into a single regression problem, whereas methods like Faster R-CNN [10] require handling both classification and regression separately. This streamlined approach makes YOLO exceptionally fast and efficient, making it suitable for real-time applications like autonomous driving and video surveillance. YOLOv8 is an evolution of the YOLOv5 [14] framework, introducing a range of enhancements in terms of architecture and developer usability. This new version offers improved speed and accuracy compared to YOLOv5 and unifies the framework for training models capable of object detection, instance segmentation, and image classification.

3.3 Classification Model

In the realm of computer vision, Convolutional Neural Networks (CNNs) have gained significant popularity for their ability to process and interpret visual data. Unlike traditional neural networks relying on matrix multiplications, CNNs employ convolution as a key operation. A CNN typically consists of three essential layers: the convolutional layer, the pooling layer, and the fully connected (FC) layer. The convolutional layer initiates the network, while the FC layer constitutes the final stage. The progressive complexity of a CNN from the convolutional layer to the FC layer allows it to systematically recognize larger segments and more intricate features within an image, ultimately leading to the identification of complete objects.

VGG 16 and 19 Model. VGG16 [19] is a variant of the CNN model which is composed of 16 layers. It gained recognition for achieving a notable top-5 test accuracy of around 92.7% on the ImageNet dataset at the time. Distinguishing itself from its predecessors, VGG16 replaces larger kernel-sized filters with a sequence of smaller 3×3 kernel-sized filters, leading to substantial improvements compared to AlexNet [22]. This design choice enhances the network's ability to capture intricate features and patterns in images. Featuring 16 layers, VGG16 excels in classifying images across 1000 object categories, encompassing a diverse range of items such as keyboards, animals, pencils, etc. Notably, the model processes images with a standard input size of 224-by-224 pixels. The architecture of VGG16 is characterized by its utilization of small 3×3 convolution filters, constituting a total of 13 convolution layers and 3 fully connected layers. The use of multiple convolution layers with small filters allows VGG16 to capture both small and large features, such as edges, textures, and shapes at different spatial scales, making it highly effective for tasks such as image recognition and analysis.

An extended version of VGG16 is VGG19, which includes 19 weight layers. It features 16 convolutional layers and three fully connected layers, similar to VGG16. Like VGG16, VGG19 utilizes 3×3 filters in its convolutional layers and 2×2 max pooling layers. In addition, VGG19 also incorporates five max-pooling layers in its architecture. This design contributes to the model's ability to capture intricate features in the input data through a hierarchical series of convolution and pooling operations. For this research, the model input was adapted to a size of $32 \times 32 \times 3$ to suit the characteristics of the input dataset. This modification allows the model to effectively process and analyze the input images in the research context.

ResNet50. The Residual Network, commonly known as ResNet [17], was introduced in 2015 and achieved remarkable success, securing first place in the ILSVRC 2015 classification competition with an impressively low error rate of 3.57%. It also emerged victorious in the 2015 ILSVRC and COCO competitions across various tasks, including ImageNet detection, ImageNet localization,

COCO detection, and COCO segmentation. The initial ResNet architecture, ResNet-34, comprised 34 weighted layers. Unlike VGGNet, which utilized 3×3 filters in each convolutional network, ResNet adopted a simpler design with fewer filters. The ResNet architecture adheres to two key design principles: each layer maintains the same number of filters based on the size of the output feature map, and if the feature map's size is halved, the number of filters is doubled to preserve temporal complexity. Numerous ResNet variations exist, differing in the number of layers while adhering to the same design principles. ResNet50 is a specific variant of the ResNet architecture, known for its deep neural network structure that consists of 50 layers. It was specifically designed to handle complex tasks and achieve high accuracy in various domains such as image classification, object detection, and semantic segmentation. For ResNet50 and higher versions, a minor modification was introduced: shortcut connections, which used to skip two layers, now skip three layers. Additionally, a 1×1 convolution layer was added. In our research, a notable modification was made by adjusting the model input to $32 \times 32 \times 3$, showcasing the flexibility of ResNet in handling diverse input sizes.

LeNet Model. The Lenet-5 [23] model, introduced in 1998, is a pioneering pre-trained model known for its simplicity and effectiveness in handwritten and machine-printed character recognition. This multilayer CNN has five layers with learnable parameters, making it a foundational milestone in the development of CNNs for visual recognition. The architecture of Lenet-5 consists of three sets of convolution layers interspersed with average pooling operations, allowing it to extract hierarchical features from input images. Two fully connected layers follow the convolution and pooling layers, enabling complex pattern recognition. The model concludes with a Softmax classifier for image categorization. In this research, we modified the Lenet-5 model to process $32 \times 32 \times 3$ RGB images with three channels for input data.

The LeNet architecture begins with a convolution layer that utilizes sixty 5×5 filters, resulting in feature maps of size $28 \times 28 \times 60$ and $24 \times 24 \times 60$. After the first convolution, a max pooling operation is applied, halving the feature map size while maintaining the number of channels. The second convolution layer uses thirty 3×3 filters, producing feature maps of dimensions $10 \times 10 \times 30$ and $8 \times 8 \times 30$. Another max pooling layer follows, reducing the feature map to $4 \times 4 \times 30$. The output is then flattened to produce 480 values. A fully connected layer with 500 neurons is employed, and the final output layer with 29 neurons corresponds to the twenty-nine classes in the dataset. This modified Lenet-5 model is designed for image classification tasks, showcasing its adaptability and continued relevance in the field.

4 Results and Analysis

In our research, we conducted experiments using Google Colab, a platform specifically designed for training models. To ensure efficient training, we utilized a

high-performance graphics card known as the Nvidia Tesla T4 GPU, which offers 16 GB of VRAM (Video Random Access Memory). This GPU allowed us to process and analyze large amounts of data effectively. To optimize our models using different techniques, we followed the default parameter settings provided in the training code, except for the batch size. We try to maximize the batch size by considering the number of parameters in our models and the available VRAM size, ensuring that it is maximum and does not exceed 32. To determine the most effective methods for the TSR task, we initially trained all models on the GTSDB and GTSRB datasets. Our objective was to select the best models for both detection and classification, which would be suitable for utilization in various applications. The performance of various models on these datasets is summarized in Tables 1 and 2.

Table 1. Performance of detection models on GTSDB dataset

Model	mAP@50	Parameters
Faster R-CNN	91.75%	144M
YOLOv7	98.50%	36.9M
YOLOv8m	99.40%	25.9M
YOLOv8n	98.10%	3.2M

Table 2. Performance of classification models on GTSRB dataset

Model	Accuracy	Parameters
LeNet	98.2%	0.4M
VGG16	97.6%	15.0M
VGG19	96.3%	20.3M
ResNet50	93.3%	24.7M

In order to update the models for detection and classification independently, we have divided the tasks into two separate components: detection and classification. The detection component focuses solely on locating and identifying traffic signs within an image. It aims to identify regions of interest that potentially contain traffic signs. By separating the detection task from classification, we can update the detection model independently without affecting the classification model. On the other hand, the classification component is responsible for predicting the specific type of traffic sign within the identified regions. It takes the regions of interest detected by the detection model and classifies them into different traffic sign categories. By decoupling the classification from the detection process, we can update the classification model independently without needing to modify the detection model. This separation allows for more flexible updates, especially in scenarios where there is a lack of data for one specific task. As shown in Table 1, our evaluation reveals that YOLOv8 demonstrates the highest mAP@50, about 99.40% for YOLOv8m. Remarkably, YOLOv8n achieves close to YOLOv8m performance while utilizing the lowest number of parameters. Notably, the mAP@50 exceeds 98.10%. Based on these findings, we have decided to select YOLOv8n as the primary detection model for our ongoing development efforts. In terms of classification, our evaluation demonstrates that LeNet achieves outstanding accuracy, with a remarkable score of 98.2% while utilizing only 0.4 million parameters. Considering its exceptional performance,

speed, and parameter efficiency, we select LeNet as our primary classification model for the next stage of our project.

Table 3. Performance of YOLOv8n and LeNet on 4 datasets

Model	Dataset	Number of classes	mAP@50	Accuracy
YOLOv8n	GSTDB	1	98.1%	-
Lenet	GTSRB	43	-	98.2%
YOLOv8n + LeNet	GSTDB + GTSRB	43	98.1%	-
YOLOv8n	GPVNS	1	94.9%	-
Lenet	ITS29	29	-	99.23%
YOLOv8n + LeNet	GPVNS + ITS29	29	90.7%	-

In the subsequent phase, we proceeded to train our detection model using the GPVNS dataset, while the classification model was trained on the ITS29 dataset. As the focus was solely on traffic sign detection, we configured the model to have a single class. Remarkably, the YOLOv8n model achieves a high mAP@50 on both the GSTDB and GPVNS datasets as shown in Table 3, achieving 98.1% and 94.9% respectively. When the YOLOv8n model is combined with LeNet, there is a possibility of a performance drop in the case of the GPVNS and ITS29 datasets. The combined model achieves a mAP@50 of only 90.7, which is lower than the original YOLOv8 model's performance of 94.9%. However, it is worth noting that the performance of the original YOLOv8n and the combined model remains mostly the same when evaluated on the GSTDB and GTSRB datasets. This suggests that the performance of the combined model is influenced by the difficulty of the dataset, which can lead to performance improvements or decreases depending on the dataset characteristics.

5 Conclusion

In summary, our research provides a comprehensive analysis of various object detection architectures applicable in the field of TSR, along with their performance comparison. We propose a combined approach using YOLOv8 and LeNet, which offers ease of maintenance and development. Additionally, we introduce two new datasets: the GoPro Viet Nam Street dataset (GPVNS) for detection tasks, and the Internet Traffic Sign 29 dataset (ITS29) for classification tasks. Our study aims to address the crucial role of Artificial Intelligence in traffic management, driven by the need for accident reduction in driverless traffic scenarios. By focusing on TSR, an essential element of this challenge, we thoroughly investigate models for both detection and classification tasks. Through our experiments, we obtain insightful results highlighting the strengths of YOLOv8 in traffic sign detection, with YOLOv8n particularly excelling in real-time processing scenarios. For the classification task, LeNet emerges as the top-performing

model, demonstrating its superiority in our evaluation. These findings emphasize the potential of AI in enhancing traffic safety, with specific models identified as robust solutions for practical applications. The knowledge gained from our research contributes to the ongoing efforts to implement effective and dependable TSR systems, as we envision a future with driverless traffic conditions. As AI continues to advance, these insights will be instrumental in developing intelligent systems that support the overarching goal of creating safer and more efficient transportation environments.

References

1. Wu, J., Zhong, L.: A new data aggregation model for intelligent transportation system. In: Construction and Urban Planning, Series Advanced Materials Research, vol. 671, pp. 2855–2859. Trans Tech Publications Ltd. (2013)
2. Hazelhoff, L., Creusen, I.M., de With, P.H.: Exploiting street-level panoramic images for large-scale automated surveying of traffic signs. Mach. Vis. Appl. **25**, 1893–1911 (2014)
3. Timofte, R., Zimmermann, K., Van Gool, L.: Multi-view traffic sign detection, recognition, and 3D localisation. Mach. Vis. Appl. **25**(3), 633–647 (2011)
4. Ai, C., Tsai, Y.C.J.: Critical assessment of an enhanced traffic sign detection method using mobile LiDAR and INS technologies. J. Transp. Eng. **141**, 5 (2014)
5. Wang, D., Zhu, J.: Fast smoothing technique with edge preservation for single image dehazing. IET Comput. Vis. **9**, 950–959 (2015)
6. Nguyen, D.T., Tran, P.-N., Phan, M.K., Dang, N.M.D.: Vietnamese traffic sign recognition using deep learning. In: Proceedings of the 2024 9th International Conference on Intelligent Information Technology. ACM (2024)
7. Ravindran, R., Santora, M.J., Faied, M., Fanaei, M.: Traffic sign identification using deep learning. In: International Conference on Computational Science and Computational Intelligence (CSCI), pp. 318–323 (2019)
8. William, M.M., et al.: Traffic signs detection and recognition system using deep learning. In: Ninth International Conference on Intelligent Computing and Information Systems (ICICIS), pp. 160–166 (2019)
9. Liu, W., et al.: SSD: single shot MultiBox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
10. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017)
11. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C.: MobileNetV2: inverted residuals and linear bottlenecks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)
12. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, Series ICML 2015, pp. 448–456. JMLR.org (2015)
13. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788 (2016)

14. Jiang, L., Liu, H., Zhu, H., Zhang, G.: Improved YOLO v5 with balanced feature pyramid and attention module for traffic sign detection. In: MATEC Web Conference, vol. 355, p. 03023 (2022). https://doi.org/10.1051/matecconf/202235503023
15. Rumelhart, D., Hinton, G., Mcclelland, J.: A general framework for parallel distributed processing. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1 (1986)
16. Girshick, R.: Fast R-CNN. In: IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448 (2015)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)
18. Hosang, J., Benenson, R., Schiele, B.: Learning non-maximum suppression. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6469–6477 (2017)
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR, vol. abs/1409.1556 (2014). https://api.semanticscholar.org/CorpusID:14124313
20. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517–6525 (2017)
21. Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M.: YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7464–7475 (2023)
22. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017). https://doi.org/10.1145/3065386
23. Kuo, C.-C.J.: Understanding convolutional neural networks with a mathematical model. J. Vis. Commun. Image Represent. **41**, 406–413 (2016). https://www.sciencedirect.com/science/article/pii/S1047320316302267