

Blended Realism: A Hybrid Dataset and Deep Learning Approach for Traffic Sign Recognition in Bangladesh

Md. Habib Ehsanul Hoque^{1*}, Dipankar Das¹

Abstract

To build robust and accurate traffic sign recognition systems in diverse and complex environments, it is necessary to use a specific hybrid dataset creation technique, which combines both real-world and simulated datasets. We present an algorithm for data generation using blending that allows for traffic sign embedding on realistic roadside backgrounds with consideration of a context-sensitive data generation technique suitable for the local roads of Bangladesh. In contrast to traditional geometry-based augmentation, our solution retains the environmental elements such as lighting, shadows, occlusion, and real-world sign placements observed by BRTA, thus considerably narrowing down the domain difference between the synthetically generated training data and the real-life application. Together with the hybrid deep learning model framework, our solution is shown to increase the generalization and robustness of the traffic sign recognition system. Our contribution to the area of intelligent traffic sign recognition involves the introduction of a context-aware blended data set creation method as well as the validation of the hybrid deep learning architectures that enable more effective traffic sign detection in real-world scenarios. The resulting dataset contains 18,750 images spanning 46 traffic sign categories, with 70% blending-based images and 30% real images. Three models YOLOv8, YOLOv8 with VGG19, and YOLOv8 with ResNet50 were evaluated under different training and testing setups, achieving F1 scores of 0.952, 0.947, and 0.957, and mAP@50–95 values of 0.660, 0.668, and 0.670, respectively. Among them, the YOLOv8-ResNet50 model achieved the highest performance. Our findings highlight the effectiveness of hybrid dataset generation in advancing traffic sign detection and recognition in practical environment.

Keywords: Bangladeshi Traffic Sign, Hybrid Model, Data Augmentation, Blended Image Generation, Hybrid Dataset.

1. Introduction

Our society desperately needs an intelligent traffic system to help our local drivers by creating an autonomous navigation system and automating applicable

*Corresponding Author email: ehsanamil@gmail.com

¹ Dept. of Information and Communication Engineering, University of Rajshahi, Bangladesh.
DOI: <https://doi.org/10.64494/JFST/v4i1/mh/2026/09/165-194>

traffic laws. A well-annotated dataset that is representative of various environments is the foundation of a successful Traffic Sign Recognition (TSR) system. However, our nation is severely lacking in people who can recognize the various types of traffic signs that are shown on highways, in rural areas, and in metropolitan areas. Because of their infrequent occurrences, uneven usage, or regional distinctiveness, many signs are either sparsely represented or nonexistent. These problems present significant challenges for deep learning model training and validation, especially for applications involving real-time detection and classification. The Bangladeshi traffic sign is grouped into four major categories regulatory, warning, informational, and additional signs each serving a critical role in ensuring road safety and guiding traffic flow. However, these signs vary widely depending on geographical location. They are often impacted by environmental conditions such as poor lighting, occlusion, worn-out materials, and deviations from BRTA (Bangladesh Road Transport Authority) standards. Recent research in Bangladesh has sought to bridge this gap by creating benchmark datasets, such as the Bangladesh Road Traffic Sign Dataset, and utilizing sophisticated deep learning architectures, such as YOLOv8 (Hoque et al., 2024), for real-time detection and classification. While these efforts have shown promise, existing methodologies often have shortcomings in several areas. Augmentation often happens in a similar dataset with minor changes in the background scene that are redundant in the visual context. Though it is a helpful technique for the expansion of the dataset, the augmentation techniques like rotation, scaling, flipping, brightness adjustment, and shearing often fail to make a diverse dataset. For this reason, deep learning models perform poorly and are trained on such data that often over fit to a limited set of conditions, making it more difficult to generalize them into real-world scenarios where different background lighting conditions, texture, shadows, or occlusions exist. Furthermore, these augmentation approaches fail to simulate natural sign degradation (e.g., blur, dirt, fading) and realistic installation settings, such as varying pole heights, angles, or partial occlusions caused by trees or vehicles.

Mismatches often occur between the training and deployment environments. Thus, a large domain gap is created. In practice, the model results in poor performance during validation. In order to overcome these limitations, we propose an approach that blends the original and generated images, which involves seamlessly integrating traffic sign images into authentic Bangladeshi roadside backgrounds. We give careful thought to maintaining the following: Accurate placement of signs at 15%, 25%, 35%, and one random position on the left, right, and center of the road based on actual BRTA-observed patterns, lighting and shadow alignment with the back-ground scene, Natural occlusions

and clutter, such as poles, pedestrians, or foliage, and degradation effects, like blur, noise, or faded edges, mimicking damaged signs. While traditional augmentation and synthetic data generation techniques often fail to maintain contextual and environmental realism, overlooking critical aspects such as realistic placement, lighting variations, occlusions, shadows, structural integration (e.g., poles and mounting), and degradation effects like blurring or fading (Li et al., 2022; Sekizawa and Nakajima, 2019), this blending method employed in this study produces more realistic training images, reduces environmental bias, and significantly enhances the model's generality to handle complex and variable real-world scenes. It enhances the diversity of both the background and the foreground elements, particularly for rare or region-specific signs that are often overlooked in conventional datasets. Most existing synthetic traffic sign datasets are developed for traffic systems in countries such as China and the United States and may not adequately reflect the characteristics of Bangladeshi roads (Møgelmoose et al., 2012; Qian et al., 2016). To address this gap, a hybrid dataset generation approach is proposed, combining real and blending-generated traffic sign images within authentic Bangladeshi roadside environments. By preserving contextual elements such as lighting conditions, shadows, occlusions, and typical sign placement patterns, the proposed dataset better represents real-world traffic scenarios.

Using this dataset, three deep learning models YOLOv8, YOLOv8-VGG19, and YOLOv8-ResNet50 are evaluated. The study investigates the effectiveness of hybrid datasets for traffic sign recognition and examines their impact on model performance and generalization (Luan et al., 2023; Redmon et al., 2016). Unlike GAN-based methods that generate entirely synthetic signs, our approach focuses on context-aware integration of real traffic sign images into realistic road scenes.

The main contributions of this work are as follows:

- a. A context-aware blending framework for integrating real traffic sign images into authentic Bangladeshi road environments.
- b. A hybrid dataset construction strategy that combines real and blending-generated images to improve dataset diversity and model generalization.
- c. A comparative evaluation of YOLOv8, YOLOv8-VGG19, and YOLOv8-ResNet50 under real, generated, and hybrid training scenarios.

2. Literature Review

Hybrid deep learning models have gained increasing attention in TSR for addressing class imbalance, data scarcity, and environmental variations. YOLO-based hybrid architectures combined with VGG19 or ResNet have shown

improved robustness and detection accuracy. Furthermore, demonstrated that GAN- and VAE-generated synthetic traffic signs can effectively improve the detection of rare sign classes (Konushin et al., 2021), emphasizing the value of synthetic data in TSR systems.

Da Silva and Fernandes (2023) developed a joint real synthetic dataset repository to strengthen the training of convolutional neural networks (CNNs), aligning datasets to improve model generalization. In essence, this repository gives researchers a hand to boost the performance of TSR systems. Horn and Houben (2020) proposed a set of realistic rendering techniques to maintain photorealism and developed a fully automated pipeline that replaces traffic signs in real-world images with synthetic renderings, enabling large-scale data augmentation. It helps to extend datasets without the need for manual annotation and therefore it can improve the detection of rare or unseen sign classes (Y. Wang et al., 2023). Yang et al., (2023) proposed a multitask learning system for traffic sign perception in real road scenes, which generates natural language descriptions of signs. This approach provides an easier and more intuitive way for people to understand the overall working process of autonomous driving (AD) systems. Shakhuro et al., 2019 designed a network for rare traffic sign recognition, leveraging synthetic training data to mitigate the problem of class imbalance.

Li et al., 2022 proposed a context-aware image synthesis technique that incorporates the semantic information of a scene to achieve realistic traffic sign placement, thereby improving recognition performance even in cluttered environments. AbuKhait, 2024 highlighted the impact of environmental conditions on road sign recognition, while Zhao et al., 2023 improved model robustness using synthetic data generation and augmentation techniques. Similarly, Ge, 2023 proposed a specialized data augmentation method to improve the usability of traffic sign recognition datasets, enabling more effective training of models when only a limited number of examples are available. Villalonga et al., 2020 demonstrated GAN-based synthetic-to-real domain adaptation to generate training samples for new, unseen traffic sign classes, and conducts extensive experiments on the Tsinghua dataset with a synthetic counterpart (SYNTHIA-TS), demonstrating that GAN-transformed synthetic images can effectively train classifiers for unknown sign classes without requiring real annotated data collection. Tabernik and Skočaj, 2020 employed a Mask R-CNN-based deep model to address the full traffic sign detection and recognition pipeline using a closed-loop training method, achieving significant improvements in overall performance. Shi et al., 2024 introduced Flexible CP, a flexible data augmentation strategy for traffic sign detection, and demonstrated its effectiveness in improving detection performance. Tabelini et al., 2022

demonstrates that training a deep detector exclusively on a fully synthetic dataset can almost match the performance of training with real images on large datasets, and when compared to training with a smaller real-image set, training with synthetic images improved accuracy by 12.25%, supporting the case for synthetic-only training pipelines. Raisa and Yasmin (2025) highlighted the crucial role of data quality for reliable recognition systems, and Shafi and Mirza, 2025 showed the impact of using pretrained deep learning architectures on the performance of image classification. Beyond synthetic data generation, recent research has explored hybrid architectures that combine YOLOv8 (Elshamy et al., 2024; Huang et al., 2023) with deep feature extractors such as VGG19 and ResNet.

Canese et al. (2022) demonstrated the effectiveness of VGG19 and ResNet for object detection due to their strong feature representation capabilities, while G. Wang et al., (2025) improved YOLOv8n and achieved a 5.7% increase in mAP@50 on the TT100K dataset. Hybrid YOLO models have shown superior accuracy and robustness compared to standalone detectors, particularly in challenging visual environments (Katsamenis et al., 2025). Furthermore, ResNet enhances semantic feature learning through residual connections, whereas VGG19 effectively captures fine-grained spatial details, making both architectures suitable for traffic sign detection under adverse conditions (He et al., 2016). These findings provide strong motivation for investigating YOLOv8-VGG19 and YOLOv8-ResNet hybrid architectures for weather-robust traffic sign recognition.

3. Methodology

The proposed framework combines real-world and blending-generated traffic sign images to construct a hybrid dataset for traffic sign detection and classification. After annotation, the dataset is used to train and evaluate YOLOv8, YOLOv8-VGG19, and YOLOv8-ResNet50 models. The overall workflow is illustrated in Figure 1.

3.1 Data Collection

In order to create a complete and representative traffic sign database for Bangladesh, photographs of a variety of road scenes have been taken ranging from highway, urban, rural, bridge, flyover, single lane, multi-lane and restricted access roads. The goal was to document as much as possible the variation in traffic signs that can be seen by drivers in the real world and to build a dataset that could be used to train and test powerful deep learning models. The final dataset is about 6000 annotated images for 46 different classes of Bangladeshi traffic signs such as regulatory, warning, mandatory, informational etc.

High resolution cameras mounted on smartphones, DSLR cameras, and vehicles were used to acquire images. The data were gathered under different environmental circumstances to make sure that the data set comprehensively represents the real road situations. Images were taken at various time of day (morning, afternoon, evening, nights) and weather conditions (sunny, cloudy, rainy, foggy) conditions. In addition to this, traffic signs were photographed from various angles, distances, and camera vantage points to add natural variations to the images, in terms of scale, orientation, illumination, and background complexity. This approach enables the proposed model to learn how to generalize to a wider range of road conditions.

The collected traffic signs intentionally leave the problems that may be encountered on the streets of Bangladesh, unlike many traffic sign datasets collected in relatively controlled conditions that are available publicly. Some of these challenges involve the smallness and distance of traffic signs, motion blur from moving vehicles, occlusion from trees, utility poles, road-side buildings and other vehicles, and the degradation of traffic signs due to fading, damage or poor maintenance. Further variations were captured to depict realistic traffic scenes, such as shadows, uneven light and dark, adverse weather conditions, cluttered backgrounds, and low visibility. The following set of images in Figure 2 and 3 are representative examples of the challenging scenarios: multiple traffic signs in a single image, traffic signs that are occluded, rain/snow, fog, and at night.

The data set also includes natural image-quality variations due to different imaging devices. The images show variations in resolution, sharpness, exposure, contrast, color balance and sensor noise, which are typical of the image properties found in ITS (Binbusayyis, 2024). These variations help to strengthen the proposed recognition framework and to make it less susceptible to any given camera hardware and acquisition conditions.

After collecting the data, each traffic sign was hand-annotated with rectangular bounding boxes in each traffic sign category. The annotations were thoroughly reviewed to guarantee accurate localization of the objects and proper classification of the classes. Pictures selected during the quality assurance process that had significant annotation inaccuracies, duplicate samples or poor visual quality were excluded. The final dataset is therefore a realistic and challenging benchmark to reflect the complexity of the road environment in Bangladesh and to create a reliable deep learning-based traffic sign recognition system for real-world applications.

3.2 Blending-Based Road Sign Embedding in Real-World Scenes

To enhance the dataset and simulate real-world conditions, a traffic sign image generation technique was developed using a placement algorithm based on computer vision that integrates a real-world road background and sign images. The generation process began by collecting a large set of road scene images from diverse locations. A trans-parent PNG image of a traffic sign was then programmatically blended into these road backgrounds.

As shown in Figure 2, each road image undergoes a region-of-interest (ROI) detection phase, where the lower half of the image is analyzed using Canny edge detection and Hough Line Transform. This helps identify suitable areas for sign placement, typically on the left side of the road where traffic signs are commonly located. A bounding region is defined within this detected area to guide realistic and consistent placement.

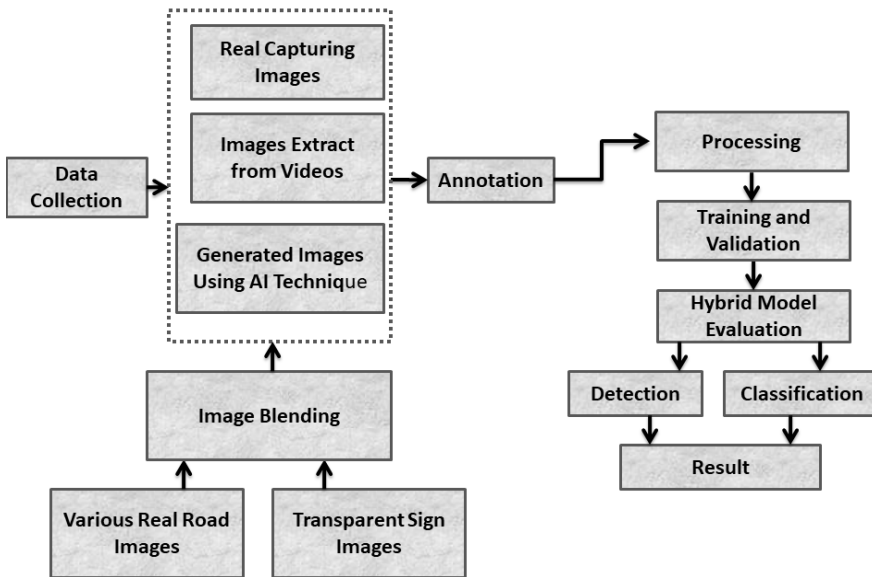


Figure 1: Overall workflow of the proposed traffic sign recognition framework, illustrating data collection, preprocessing, synthetic image generation, dataset augmentation, model training, and performance evaluation.

The sign image is then dynamically resized in each instance using randomized shrink and zoom factors to simulate scale variation. The resized sign is carefully over-laid on the selected road region, ensuring that its position does not overlap with the main drivable area. After compositing, each final image is resized to a

consistent resolution to maintain uniformity across the dataset. To create a variety of realistic training images for traffic sign detection, a systematic algorithm was developed for placing road signs at different positions along the road. This algorithm overlays traffic sign images onto real-world road backgrounds by detecting appropriate regions for placement. The method ensures that each generated image maintains realism by considering road geometry, sign scaling, and plausible spatial location near the road. Edge detection and line extraction techniques are applied to localize the road area, and the placement logic ensures the sign appears on the left side of the road without overlapping the main drivable region. The details of the process are outlined in Algorithm 1, which was used to generate over augmented road images for training.



Figure 2: A representative sample of the collected real-world data set to highlight multi-sign scenes and challenging traffic sign instances in diverse environments, such as occlusion, low visibility, and camera-quality changes.

A few examples of the resulting generated images, showing traffic signs embedded naturally into diverse road scenes, are illustrated in Figure 5. This

technique resulted in high quality blending images that significantly enriched the dataset, particularly for underrepresented sign classes, helping to improve model robustness and generalization.



Figure 3: The examples of traffic signs captured at various scales from the camera (from ~40 m to ~5 m), which reflects the scale difference in real driving environments.

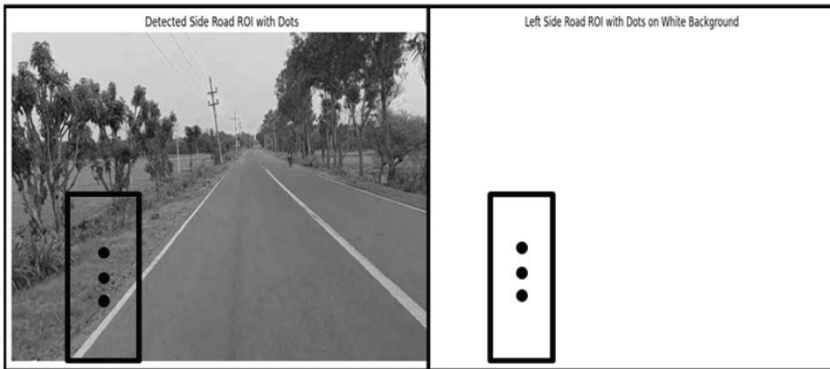


Figure 4: Background image with the detected region of the road and the region of interest (ROI) for placing the traffic signs. Lower half of image analyzed for road boundaries to direct placement of signs.

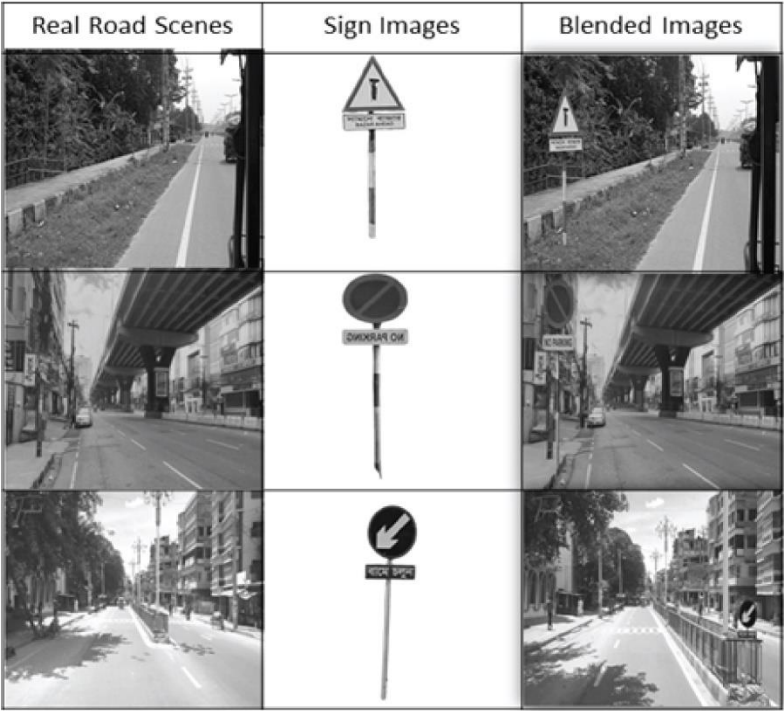


Figure 5: Examples of blending-based generated road images where traffic signs are placed in contextually realistic positions using road-aware placement techniques

3.2.1 Challenges and Gaps in Data Augmentation Strategy

The images generated were then subjected to geometric augmentation to add variety of scale, orientation, and perspective to the dataset. These transformations make the model more robust and make synthesized samples more realistic (Alsiyeu and Duisebekov, 2024). However, conventional augmentation techniques face several challenges such as lighting variation, unrealistic object boundaries, lack of occlusion modeling, and limited diversity of background scenes, which can lead to a domain gap between these augmentations and real-world scenes. To tackle these problems, image generation process based on blending is applied in the proposed method to enhance the realism of the dataset and the generalization of the model by adding contextual placement and realistic visual integration.

3.2.2 Blended Image Generation versus Traditional Augmentation

For generating diversity in dataset, conventional augmentation and blending-based image generation were used. The proposed blending method differs from the conventional augmentation by rendering traffic signs within a real road context while retaining other contextual elements like lighting, texture and perspective. This creates more realistic training examples, and enhances the generalizability of the model. Compositing process is discussed in Equation (1).

$$I(x, y) = \alpha(x, y) \cdot T(S(x, y)) + (1 - \alpha(x, y)) \cdot B(x, y) \quad (1)$$

Where

$S(x, y)$ = image of the traffic sign,

$B(x, y)$ = road image with the background,

T = transformation operator (scaling, rotation, shearing) and

$\alpha(x, y)$ = blending coefficient

The blending coefficient $\alpha \in [0, 1]$ is extracted from the alpha channel of the traffic sign image that is transparent in PNG format, and renormalized in the range $[0, 1]$. $\alpha(x, y)$ differs at each pixel based on the transparency of each individual sign pixel and is not fixed. In particular, when $\alpha = 1$, the sign regions will be fully opaque, while background regions will be fully transparent when $\alpha = 0$, and will be smoothly transitioning between sign and background regions in between. This alpha blending at the pixel level provides realistic background incorporation and achieves better visual fidelity of traffic signs, which in turn enhances the quality and realism of the created training samples.

Algorithm 1 Road Sign Placement Algorithm

Require: Road images $T \in \{R_i\}$, sign images $S \in \{S_j\}$, output directory D

Ensure: Generated images $G = \{G_1, G_2, \dots, G_m\}$

- 1: **for** each sign image $S_j \in S$ **do**
- 2: Load sign image S_j
- 3: **for** $i = 1$ to n **do**
- 4: Load road image $R_i \in T$
- 5: Create output directory D
- 6: Convert R_i to grayscale and apply Canny edge detection
- 7: Extract road lines using Hough Line Transform
- 8: Define lower half road area as mask A_i
- 9: Compute bounding box $A_i = (X_{\min}, X_{\max}, Y_{\min}, Y_{\max})$

- 10: Calculate scale factor based on distance of placement:
 $d = \text{distance of placement}$
 $f = f_{\text{base}} \cdot (1 + \delta)$ where $f_{\text{base}} \sim U(0.1, 0.3)$, $\delta \sim U(0.5, 1.0)$
 $f = k \cdot d$ (where k is a scaling constant)
- 11: Resize S_j with scale factor f
- 12: Calculate sign placement coordinates:
 $x = \max(10, X_{\min} - w' - 20)$
 $y \sim U(Y_{\min} + 50, Y_{\max} - h' - 50)$
- 13: Overlay resized sign S'_j on R_i at (x, y) to obtain G_i
- 14: Save G_i in directory D
- 15: **end for**
- 16: **end for**

3.2.3 Comparative Insights

While traditional augmentation can produce geometry diversity, it lacks realism; blending-based image generation on the other hand can generate context-aware samples that can more closely represent real world. These methods complement each other by offering a broader range of datasets, narrower domain gap and better model robustness. Table 1 presents a comparison of the two.

Table 1: Comparison the traditional augmentation and the image generation using blending method

Aspect	Traditional Augmentation	Blending-Based Generation
Scale/Orientation Variation	High	High
Lighting Consistency	Low	High
Occlusion Handling	Absent	Partially Realistic
Background Diversity	Low (static)	High (real-world scenes)
Computational Cost	Low	Moderate
Domain Gap Reduction	Limited	Significant

3.2.4 Implementation Details of the Blending Pipeline

The blending pipeline contains five major processing stages: (1) preprocessing road images, (2) region-of-interest detection, (3) traffic sign scaling and placement, (4) alpha blending, and (5) generation of annotations. Images from the roads were first resized and grayscale to extract edges from them using the

Canny operator. The Hough Line Transform was then used to detect the road boundaries and calculate the possible regions for the placement of roads. The images for the traffic signs were resized as per the estimated distance of placement and then pasted in the picked area using alpha compositing. Lastly, the bounding-box annotations were created and checked prior to the exportation of the dataset in the YOLO format (Ali and Zhang, 2024).

3.3 YOLOv8 Architecture for Real-Time Traffic Sign Detection

The YOLOv8 architecture for real-time traffic sign detection framework that consists of a CSP-based backbone, an FPN-PAN neck, and an anchor-free detection head: YOLOv8. It has a simple structure, which also ensures high accuracy of the identification of traffic signs of various sizes with minimal computational complexity as illustrated in Figure 6.

3.3.1 Model Variants and Flexibility

Performs multiple tasks such as object detection, segmentation, classification and pose estimation in the same architecture. There are a number of variants, such as YOLOv8n (nano), s (small), m (medium), l (large) and x (extra-large) which trade speed for accuracy based on available hardware.

3.3.2 Deployment Support

The model can be deployed into popular frameworks such as ONNX, TensorRT and CoreML, which is ideal for edge devices and mobile platforms. The deployment options enable real-time processing when needed in traffic-aware systems, especially in resource constrained environments.

3.4 Traffic Sign Detection Using the Hybrid YOLOv8-VGG19 Model

The proposed YOLOv8–VGG19 framework is a sequential two-stages inference approach, which is different from feature fusion and end-to-end joint optimization. YOLOv8 was initially employed as an object detector to identify traffic signs in the input image. The traffic signs regions detected are then cropped and passed to the VGG19 classifier for the final prediction of the traffic sign category. Thus, detection and classification are carried out in two separate but sequential steps. The overall architecture of traffic sign recognition framework for detection and classification is shown in Figure 7.

YOLOv8 has built-in classification, but its main advantage is that it is fast and accurate for locating the objects. However, for more fine-grained recognition tasks like traffic sign classification, where signs can be structurally similar, with similar shapes and colors, or slightly blurred, occluded, or low in resolution, YOLOv8 sometimes fails, especially in difficult scenarios. We overcome this by

using a dedicated classifier: VGG19. As a more complicated convolutional network, VGG19 has proven to be excellent at extracting discriminative feature information that increases the accuracy of recognition, and it is suitable for distinguishing similar traffic signs. The hybrid model has two consecutive steps:

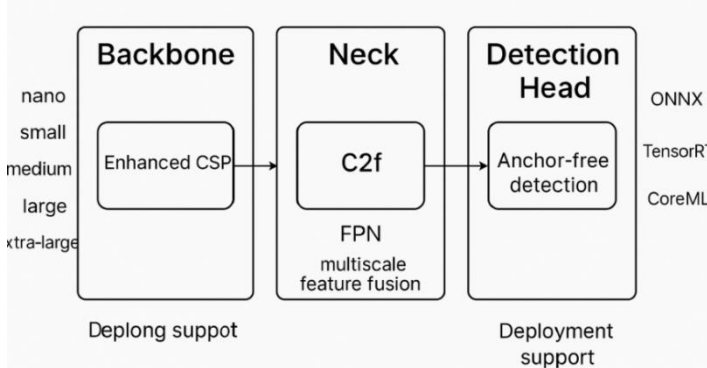


Figure 6: Architecture of YOLOv8 illustrating the backbone, neck, and anchor-free detection head designed for real-time traffic sign detection

Stage 1: Detection (YOLOv8): The input image I is processed by YOLOv8, which predicts bounding boxes for traffic signs. Each bounding box is denoted as

$$B_i = (x_i, y_i, w_i, h_i), i = 1, 2, \dots, N, \quad (2)$$

where N = the total number of detected signs and B_i =bounding box.

Stage 2: Classification (VGG19): For each bounding box B_i , the corresponding region of interest (ROI) R_i is cropped and resized to 224×224 pixels, the input size required by VGG19. As shown in Equation 2, the network maps each cropped region R_i to a probability distribution over classes using VGG19 and the softmax function.

$$P(C_i | R_i) = \text{Softmax}(\text{VGG19}(R_i)) \quad (3)$$

where, C_i = predicted traffic sign class, R_i = cropped region

The detection classification pipeline is summarized in Equation 3, where YOLOv8 generates bounding boxes and each cropped region R_i is classified into a class C_i .

$$\begin{aligned} \{B_1, B_2, \dots, B_N\} &= \text{YOLOv8}(I), \\ C_i &= \arg \max_{c \in C} P(C_i = c | R_i), R_i = \text{Crop}(I, B_i), \end{aligned} \quad (4)$$

where C denotes the set of all traffic sign classes. In this hybrid architecture, YOLOv8 acts as the detector, performing object localization in real time, while VGG19 functions as the classifier, leveraging its deep convolutional layers for robust recognition of traffic sign categories. The VGG19 architecture, shown in Figure 8, consists of 19 weight layers, including 16 convolutional layers followed by 3 fully connected layers. Each convolutional block employs 3×3 filters with ReLU activation, and max-pooling layers progressively reduce spatial dimensions while preserving essential features.

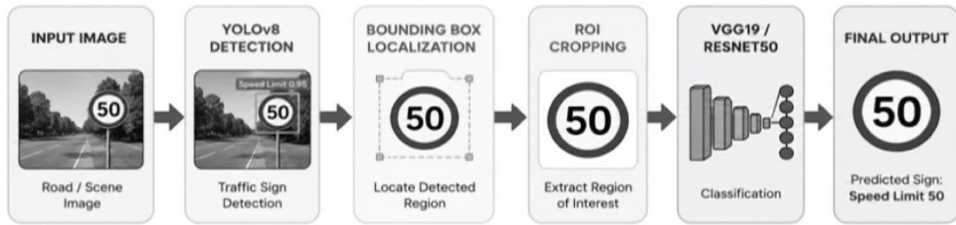


Figure 7: The overall diagram of the proposed architecture for traffic sign recognition with YOLOv8-VGG19/ResNet50 for detection and classification

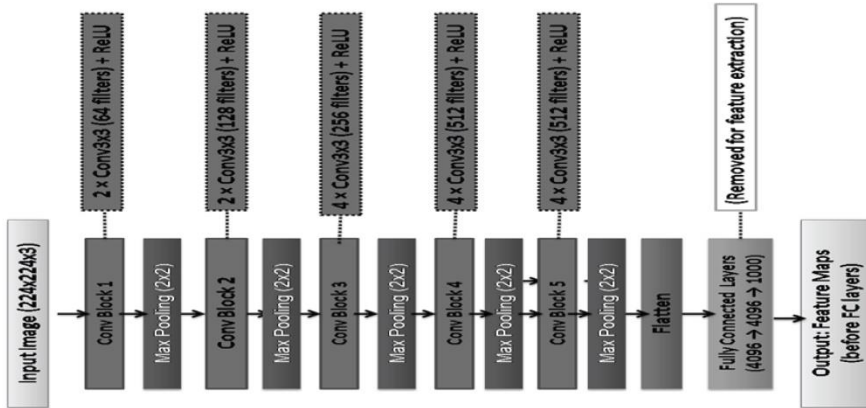


Figure 8: Convolutional blocks for feature extraction in the VGG19 framework used in the hybrid YOLOv8-VGG19 framework for traffic sign classification

The model consists of 16 convolutional layers and 3 fully connected layers with ReLU activation functions and with max-pooling to represent the hierarchical spatial features. VGG19 is designed to be very deep and straightforward, with many layers of uniform convolutional and pooling operations, making it well suited for fine-grained classification problems. It adds depth to the channels in each layer (from 64 to 512 filters) to capture more complex spatial patterns to differentiate between the traffic signs that vary subtly. The YOLOv8-VGG19

hybrid system uses real-time detection and VGG19 for discriminative classification, resulting in high accuracy rates for the detection and recognition of various Bangladeshi traffic signs in different environments. This modular design ensures that the misclassification rate due to visual similarity is minimized, that it becomes more robust in the presence of occlusion and/or blur, and that overall detection-classification performance is improved.

3.5 Traffic Sign Detection Using a Hybrid YOLOv8-ResNet50 Model

The YOLOv8-ResNet50 framework does not operate on the basis of feature fusion or joint optimization, but uses a two-stage architecture that includes the detection of traffic signs by the YOLOv8 network and the classification of the detected regions by the ResNet50 network. The YOLOv8-ResNet50 framework is based on two-stage strategy, where the YOLOv8 network is used to detect traffic signs and the ResNet50 network is used to classify the traffic signs regions. The ResNet50 architecture as shown in Figure 9 is comprised of 50 layers that are grouped into residual blocks containing convolutional operations and identity/projection shortcuts. This combination can enhance the recognition accuracy by efficiently localizing and extracting deep features, especially in complex scenarios.

$$R_i = \text{Crop}(I, B_i)$$

is cropped and resized to the input size required by ResNet50. The network extracts deep feature representations F_i , which are then mapped to class probabilities as follows:

$$\begin{aligned} F_i &= \text{ResNet50}(R_i), \\ P(C_i | R_i) &= \text{Softmax}(F_i), \\ C_i &= \arg \max_{c \in C} P(C_i = c | R_i), \end{aligned} \tag{5}$$

As shown in Equation 4, each cropped region R_i is processed by ResNet50 to extract deep feature representations F_i , which are then converted into class probabilities and finally assigned a predicted class C_i .

YOLOv8 is used for traffic sign localization, while ResNet50 provides deep feature-based classification. This combination can enhance recognition accuracy by efficiently localizing and extracting deep features, especially in complex scenarios.

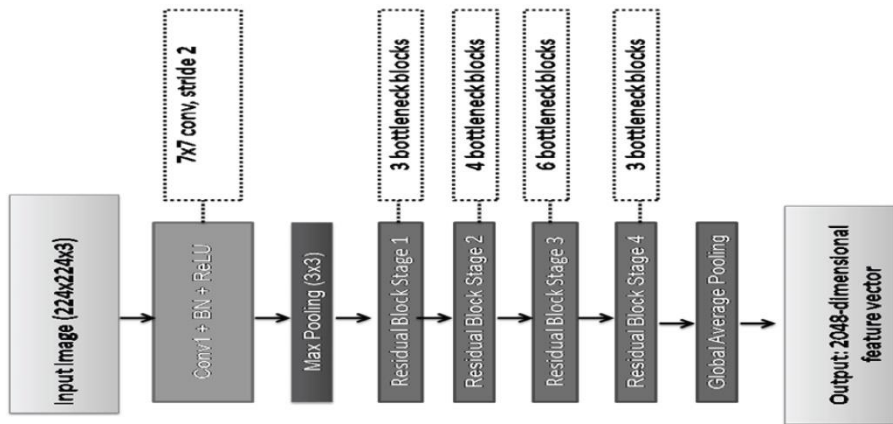


Figure 9: ResNet50 architecture for traffic sign classification using deep residual learning for robust feature extraction and efficient training

3.6 Justification of Model Selection

The models YOLOv8, VGG19, and ResNet50 were selected for their capabilities in real-time detection efficiency and feature extraction and classification. The hybrid architecture allows for a comparison of accuracy and robustness to varying traffic and environmental conditions.

3.7 Time Complexity of the Models

YOLOv8 provides efficient real-time detection and VGG19 and ResNet50 provide better feature representation with more computation. The main aim of the present research was not to compare the statistical performance of the proposed hybrid dataset to a specific model but to assess the performance of the proposed hybrid dataset in various deep learning architectures. As a consequence, the representative detection frameworks YOLOv8, YOLOv8-VGG19, and YOLOv8-ResNet50 were used to evaluate the general applicability of the proposed dataset. YOLOv8-ResNet50 has a significant performance balance of accuracy and computation. The detailed comparison shown in Table 2.

Table 2: Model Comparison

Model	Params (M)	Latency (ms)	Cost
YOLOv8	7–11	10–15	Low
YOLOv8-VGG19	144	25–35	High
YOLOv8-ResNet50	23–25	18–22	Moderate

4. Result and Discussion

The experimental outcomes presented for the three mentioned architectures (YOLOv8, YOLOv8-VGG19, and YOLOv8-ResNet50) show the advantages of training on the mixed real and synthetic traffic sign images. The other models achieved a better performance on the hybrid dataset with YOLOv8 baseline model achieving the highest precision of 0.955, recall of 0.950, F1 score of 0.952, mAP@50 of 0.968, and mAP@50–95 of 0.660. The hybrid model YOLOv8-VGG19 also made significant improvements with the highest score obtained on the hybrid data set in terms of precision (0.950), recall (0.945), F1 score (0.947), mAP@50 (0.969), and mAP@50–95 (0.668). The hybrid model of YOLOv8-ResNet50 model outperformed other models with precision of 0.960, recall of 0.955, F1 score of 0.957, mAP@50 of 0.974, and mAP@50–95 of 0.670. This setup consistently out-performed the baseline YOLOv8 and the YOLOv8-VGG19 model, especially under the difficult conditions of different sign types, occlusion, and low visibility. It is important to note that the reported performance was computed on data sets which include both single-sign and multi-sign traffic scenes. Thus, these evaluation results are based on the model performance under realistic traffic scenario in which different traffic signs could be present in the same image. The hybrid training method enhanced the generalization ability across domain boundaries and helped to achieve balance class representation in cases where some signs were rarely seen. Moreover, the ResNet50 back bone was found to be a good feature extractor for achieving classification of 46 different classes of the Bangladeshi Traffic signs. The results confirm the value of the blended data generation with the use of sophisticated hybrid architecture in the context of real-time and accurate traffic sign recognition systems. Figure 11 depicts the performance comparison in all three models for Hybrid/Hybrid configuration.

4.1 Dataset Composition and Experimental Setup

In the final dataset, there are 18,750 labeled traffic sign images belonging to 46 different classes, which is a mixed dataset of real and Blending images. The images in this dataset are obtained by taking 5500 images (about 30%) from the road; the remaining 13,250 images (about 70%) are generated by a controlled blending process with various types of road background images. Imbalanced class distribution across the data-set is achieved in each class with around 120 real images and 280 generated images.

The hybrid dataset was used in three experimental setups: (i) real image for training and test, (ii) generated image for training and test and (iii) hybrid dataset (real image + generated image) for training and test. To ensure a fair comparison

and a good evaluation, we kept a fixed ratio of 78% for training, 12% for validation, and 10% for testing in all experimental cases. Once images were generated, all images were annotated in Make Sense AI. Bounding boxes were manually checked for correct localization of traffic signs. The annotations were then exported in the format of YOLO, which is used for training and testing the model. The distribution of images across the real, generated, and hybrid datasets is presented in Table 3.

Table 3: Image distribution across training, validation, and testing sets

Type	Total	Train (78%)	Val (12%)	Test (10%)
Real	5,500	4,290	660	550
Generated	13,250	10,335	1,590	1,325
Hybrid	18,750	14,625	2,250	1,875

4.2 Model Training

In this study, we experimented with three different model configurations:

- The standalone YOLOv8 model,
- Hybrid model combining YOLOv8 with VGG19 and YOLOv8 with ResNet50.

Each model was evaluated across multiple training and testing dataset combinations to assess generalization capability and robustness. For all three models, training and evaluation were conducted using three dataset categories: *Generated only*, *Real only*, and a *Hybrid (Generated + Real)* dataset.

Each of these was paired with the same three categories during testing. Table 4 summarizes the train–test combinations used in the experiments.

Table 4: Training and testing dataset combinations used for each model

Training Data	Test: Generated	Test: Real	Test: Hybrid
Generated	✓	✓	✓
Real	✓	✓	✓
Hybrid	✓	✓	✓

This experimental setup was applied uniformly to the standalone YOLOv8, YOLOv8-VGG19, and YOLOv8-ResNet50 models. The goal was to evaluate model behavior across domain variations and to analyze how well each architecture could adapt to synthetic, real-world, and mixed data scenarios.

4.2.1 Hyperparameter Settings

The training of all models was performed with the same set of hyper parameters for each of the models to make the models more comparable. The training was performed for 100 epochs with the AdamW optimizer and initial learning rate of 0.0002. All images were resized to 640 x 640 pixels prior to training. The other hyperparameters are summarized below in Table 5.

Table 5: Training Hyperparameters Used for YOLOv8, YOLOv8-VGG19, and YOLOv8-ResNet50 Models

Hyperparameter	YOLOv8	YOLOv8 - VGG19	YOLOv8 - ResNet50
Backbone	YOLOv8	YOLOv8 + VGG19	YOLOv8 + ResNet50
Input Resolution	640 × 640	640 × 640	640 × 640
Batch Size	16	16	16
Optimizer	AdamW	AdamW	AdamW
Learning Rate	0.0002	0.0002	0.0002
Weight Decay	0.0005	0.0005	0.0005
IoU Threshold	0.70	0.70	0.70
Epochs	100	100	100
Data Augmentation	Mosaic, Scaling, Translation, HorizontalFlip, HSV Augmentation	Same	Same
Hardware Platform	NVIDIA Tesla P100 (16 GB)	Same	Same

4.2.2 Model Accuracy Measurement

To test the accuracy and robustness of the models, five performance metrics were calculated: Precision, Recall, F1 Score, mean Average Precision at IoU 0.50 (mAP@50) and mean Average Precision at IoU thresholds ranging from 0.50 to 0.95 (mAP@50–95). Precision is the percentage of the traffic signs correctly detected in all predicted traffic signs;

$$Precision = \frac{TP}{(TP+FP)} \quad (6)$$

where TP denotes True Positives and FP denotes False Positives.

Recall measures the proportion of actual traffic signs that are correctly detected by the model and is defined as:

$$Recall = \frac{TP}{(TP+FN)} \quad (7)$$

where FN denotes False Negatives.

For the F1 score, the Precision and Recall values are balanced and is calculated as:

$$\text{F1 score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (8)$$

For an overall detection assessment, Mean Average Precision (mAP) was used in addition to these metrics. mAP@50 is the mean Average Precision at IoU threshold of 0.50 and mAP@50–95 is the average mAP over IoU thresholds from 0.50 to 0.95 with a step size of 0.05. These metrics give a thorough evaluation of localization accuracy, classification performance and model robustness in varied detection situations. These metrics were measured for each model across multiple training and testing configurations, including training on generated data, real data, and hybrid data (a combination of both), as summarized in Table 6.

Table 6: Performance evaluation across training and testing configurations for YOLOv8, YOLOv8-VGG19, and YOLOv8-ResNet50

(a) YOLOv8 Model Performance

Train/Test Data	Precision	Recall	F1 Score	mAP@50	mAP@50–95
Hybrid / Hybrid	0.955	0.950	0.952	0.968	0.660
Hybrid / Real	0.940	0.932	0.936	0.965	0.645
Hybrid / Generated	0.960	0.961	0.960	0.970	0.740
Real / Hybrid	0.905	0.890	0.896	0.950	0.603
Real / Real	0.918	0.901	0.906	0.955	0.615
Real / Generated	0.885	0.878	0.880	0.940	0.589
Generated / Hybrid	0.920	0.912	0.915	0.967	0.610
Generated / Real	0.919	0.908	0.910	0.964	0.603
Generated / Generated	0.992	0.994	0.993	0.991	0.788

(b) YOLOv8-VGG19 Hybrid Model Performance

Train/Test Data	Precision	Recall	F1 Score	mAP@50	mAP@50–95
Hybrid / Hybrid	0.950	0.945	0.947	0.969	0.668
Hybrid / Real	0.945	0.950	0.947	0.970	0.665
Hybrid / Generated	0.970	0.965	0.968	0.973	0.745

Real / Hybrid	0.935	0.925	0.930	0.960	0.640
Real / Real	0.940	0.933	0.930	0.968	0.657
Real / Generated	0.900	0.880	0.890	0.942	0.610
Generated / Hybrid	0.925	0.930	0.928	0.963	0.635
Generated / Real	0.930	0.947	0.936	0.972	0.650
Generated / Generated	0.981	0.982	0.981	0.984	0.730

(c) YOLOv8-ResNet50 Hybrid Model Performance

Train/Test Data	Precision	Recall	F1 Score	mAP@5 0	mAP@ 50–95
Hybrid / Hybrid	0.960	0.955	0.957	0.974	0.670
Hybrid / Real	0.945	0.950	0.947	0.972	0.668
Hybrid / Generated	0.980	0.976	0.978	0.985	0.770
Real / Hybrid	0.940	0.933	0.936	0.965	0.648
Real / Real	0.938	0.929	0.933	0.966	0.652
Real / Generated	0.905	0.892	0.898	0.948	0.610
Generated / Hybrid	0.925	0.930	0.927	0.965	0.640
Generated / Real	0.915	0.928	0.921	0.961	0.630
Generated / Generated	0.995	0.992	0.993	0.990	0.780

4.3 Summary of Model Performance

Experimental results show the superiority of the proposed hybrid dataset throughout all the evaluated architectures. The YOLOv8-ResNet50 model outperforms the other two models in terms of precision, recall, F1-score, and mAP. YOLOv8-VGG19 further demonstrated its benefits over the baseline YOLOv8, which indicates that its deep CNN backbones provided better feature extraction. The result of the hybrid dataset demonstrates that the hybrid can achieve better performance than the real image, which means that using both real and generated images can promote the generalization, alleviate the class imbalance problem and enhance the robustness of the model under difficult conditions like occlusion and low visibility. In summary, it is demonstrated that the use of blending data generation and hybrid deep learning architecture is an effective solution in real world traffic sign recognition.

4.3.1 Comparative Performance of YOLOv8 and Hybrid Models

It is worth noting that, as can be seen in Figure 11, the F1-score analysis across different training and testing configurations reveals important insights into model generalization. Under the Trained on Generated, Tested on Generated (TGTG) configuration, all models achieved near-perfect performance, with YOLOv8, YOLOv8-VGG19, and YOLOv8-ResNet50 obtaining F1-scores of 0.9928, 0.981, and 0.993, respectively.

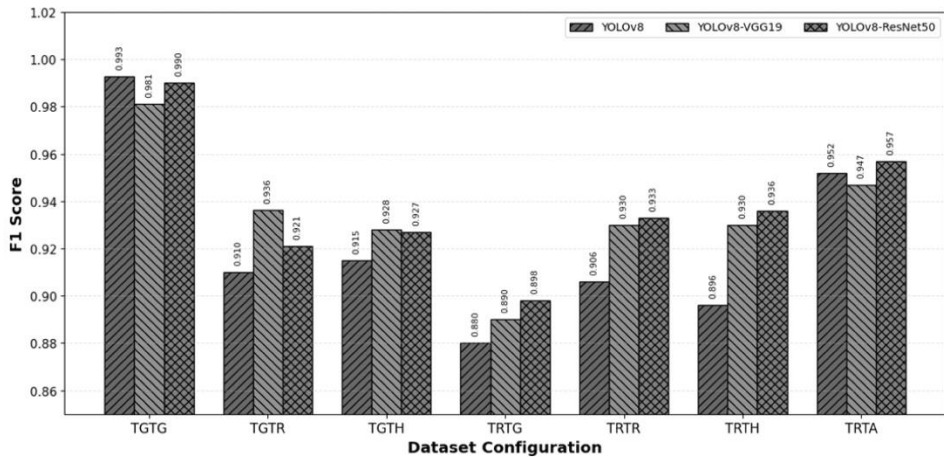


Figure 11: F1-score comparison of YOLOv8, YOLOv8-VGG19, and YOLOv8-ResNet50 across all training and testing combinations. Hybrid training improves accuracy across models, with YOLOv8-ResNet50 achieving the highest overall performance

However, performance declined under the Trained on Generated, Tested on Real (TGTR) setting due to domain differences between synthetic and real-world images. In this scenario, YOLOv8-VGG19 achieved the highest F1-score (0.9362), followed by YOLOv8-ResNet50 (0.921) and YOLOv8 (0.910), indicating improved cross-domain generalization through deeper feature extraction. Similar trends were observed for the Trained on Generated, Tested on Hybrid (TGTH) configuration, where all models benefited from partial exposure to generated samples, achieving F1-scores above 0.91.

For the reverse-domain evaluation, Trained on Real, Tested on Generated (TRTG), all models experienced performance degradation, highlighting the challenges associated with synthetic domain adaptation. Nevertheless, YOLOv8-ResNet50 achieved the highest F1-score (0.898), followed by YOLOv8-VGG19 (0.890) and YOLOv8 (0.880).

Under the Trained on Real, Tested on Real (TRTR) configuration, the models demonstrated stable performance, with YOLOv8-ResNet50 again achieving the highest F1-score (0.933). Similarly, in the Trained on Real, Tested on Hybrid (TRTH) setting, YOLOv8-ResNet50 maintained the best performance (0.936), indicating strong robustness to mixed-domain data.

Hybrid offers the best balanced real-world generalization, Tested on Hybrid (TRTH) configuration, where the combination of real and generated data improved both generalization and robustness. In this setting, YOLOv8, YOLOv8-VGG19, and YOLOv8-ResNet50 achieved F1-scores of 0.952, 0.947, and 0.957, respectively, with YOLOv8-ResNet50 also achieving the highest mAP@50–95 of 0.670.

These findings demonstrate that hybrid training effectively bridges the domain gap and provides the most reliable performance across diverse traffic sign recognition scenarios

4.5 Analysis of Blended Image Placement Errors

In our dataset generation pipeline, we used a blending-based approach to place traffic signs into real-world road scenes. Out of 500 generated images, approximately 300 images (60%) were placed accurately on the roadsides. However, the remaining 200 images (40%) exhibited placement errors due to various challenges, as illustrated in Figure 12. The main reasons for incorrect or inaccurate placement include:

Improper Roadside Detection: Sharp curves or irregular road edges often cause the algorithm to misidentify the correct placement zone. When traffic signs are blended onto these regions, the resulting images appear unrealistic or misaligned. Shown in Figure 12a.

Obstructions in the Scene: Roads with multiple objects such as trees, walls, poles, or potholes interfere with the blending process. The algorithm may overlap traffic signs with these objects, producing unrealistic placements. Shown in Figure 12b.

Scaling and Zooming Issues: Incorrect estimation of traffic sign size relative to the scene can cause signs to appear too large or too small, particularly on roads with variable widths or perspective distortions.

Lighting and Perspective Mismatch: Differences in lighting, shadows, and camera perspective between the original scene and the traffic sign image reduce visual realism, making placements appear obviously incorrect.

Texture and Occlusion Handling: Improper blending when the background has complex textures or partial occlusions can create unnatural integration of the traffic sign into the scene Shown in Figure 12c.

We are currently working on strategies to mitigate these issues, including improving roadside detection algorithms, adaptive scaling and perspective correction, and occlusion-aware blending. The analysis indicates that while blending-based image generation is effective for augmenting data, careful attention to scene geometry, object detection, and size scaling is necessary to ensure realistic placement and improve downstream model training.

No.	Real Road Scenes	Sign Images	Incorrectly Placed Images
(a)			
(b)			
(c)			

Figure 12: Failure cases of the proposed traffic sign blending algorithm: (a) incorrect roadside detection on curved roads; (b) overlapping of traffic signs with roadside objects; and (c) unrealistic blending caused by texture inconsistencies and occlusions.

5. Conclusion

In this research, a comprehensive hybrid approach for recognizing traffic signs in Bangladeshi roads is proposed based on the YOLOv8, YOLOv8-VGG19 and YOLOv8-ResNet50 models. The hybrid dataset consists of real and blended images and is composed of 46 traffic sign classes. Experimental results showed that hybrid training provided the most consistent cross-domain generalization. In the models evaluated, YOLOv8-ResNet50 achieved the significant performance with F1-score of 0.957 and mAP@50–95 of 0.670, and showed that the use of data diversity and deep feature extraction is beneficial.

5.1 Future Works

Future work will concentrate on improving contextual understanding by integrating scene-aware validation mechanisms that can validate the semantic consistency between traffic signs and their surrounding scenes. More investigations will cover comparing the proposed dataset with other state-of-the-art detectors such as Faster R-CNN, RetinaNet, EfficientDet, YOLOv10, and transformer-based ones. Moreover, future research will investigate the impact of camera-specific parameters and the effects of sensor noise, detailed ablation analysis of augmentation and blending strategies and statistical validation through repeated experiments and applying confidence interval analysis, to evaluate the reliability of the observed results further.

Acknowledgment

The authors would like to express their sincere appreciation to the individuals and institutions that provided valuable support and resources during the course of this research.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

The dataset is part of an ongoing academic degree-oriented research project and is currently being prepared for public release. It will be made publicly available through an appropriate data repository upon completion of the research and final validation.

Funding

This research received no external funding.

AI Declaration

During the preparation of this work, the authors used ChatGPT to improve the grammar and readability of the manuscript. The authors reviewed and edited the content as needed and take full responsibility for the final published version.

References

- AbuKhait, J. (2024). U.S. road sign detection and visibility estimation using artificial intelligence techniques. *International Journal of Advanced Computer Science and Applications*, 15(1), 604–612. <https://doi.org/10.14569/IJACSA.2024.0150159>
- Ali, M. L., & Zhang, Z. (2024). The YOLO framework: A comprehensive review of evolution, applications, and benchmarks in object detection. *Computers*, 13(12), Article 336. <https://doi.org/10.3390/computers13120336>
- Alsieyue, U., & Duisebekov, Z. (2024). Enhancing traffic sign recognition with tailored data augmentation: Addressing class imbalance and instance scarcity. *arXiv*. <https://arxiv.org/abs/2406.03576>
- Binbusayyis, A. (2024). Hybrid VGG19 and 2D-CNN for intrusion detection in the fog-cloud environment. *Expert Systems with Applications*, 238, Article 121758. <https://doi.org/10.1016/j.eswa.2023.121758>
- Canese, L., Cardarilli, G. C., Di Nunzio, L., Fazzolari, R., Famil Ghadakchi, H., Re, M., & Spanò, S. (2022). Sensing and detection of traffic signs using CNNs: An assessment on their performance. *Sensors*, 22(22), Article 8830. <https://doi.org/10.3390/s22228830>
- Da Silva, D. L., & Fernandes, A. R. (2023). Traffic sign repositories: Bridging the gap between real and synthetic data. In A. Fred, C. Sansone, O. Gusikhin, & K. Madani (Eds.), *Deep learning theory and applications* (Vol. 1858, pp. 56–77). Springer. https://doi.org/10.1007/978-3-031-37317-6_4
- Elshamy, M. R., Emara, H. M., Shoaib, M. R., & Badawy, A.-H. A. (2024). PYOLOv8: Efficient and accurate real-time detection of distracted driving. In *2024 IEEE High Performance Extreme Computing Conference (HPEC)* (pp. 1–6). IEEE. <https://doi.org/10.1109/HPEC62836.2024.10938429>

- Ge, J. (2023). Traffic sign recognition dataset and data augmentation. *arXiv*. <https://arxiv.org/abs/2303.18037>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778). IEEE.
- Hoque, M. H., Mahbub, M., Bin Md Tomari, M. R., Bashar, R., Rashed, M. G., & Das, D. (2024). Real-time detection of diverse Bangladeshi traffic signs using YOLOv8. In *2024 IEEE International Conference on Future Machine Learning and Data Science (FMLDS)* (pp. 203–208). IEEE. <https://doi.org/10.1109/FMLDS63805.2024.00046>
- Horn, D., & Houben, S. (2020). Fully automated traffic sign substitution in real world images for large-scale data augmentation. In *2020 IEEE Intelligent Vehicles Symposium (IV)* (pp. 465–471). IEEE. <https://doi.org/10.1109/IV47402.2020.9304547>
- Huang, Z., Li, L., Krizek, G., & Sun, L. (2023). Research on traffic sign detection based on improved YOLOv8. *Journal of Computer and Communications*, 11, 226–232. <https://doi.org/10.4236/jcc.2023.117014>
- Katsamenis, I., Bakalos, N., Lappas, A., Protopapadakis, E., Martín-Portugués Montoliu, C., Doulamis, A., Doulamis, N., Rallis, I., & Kalogeras, D. (2025). DORIE: Dataset of road infrastructure elements—A benchmark of YOLO architectures for real-time patrol vehicle monitoring. *Sensors*, 25(21), Article 6653. <https://doi.org/10.3390/s25216653>
- Konushin, A., Faizov, B., & Shakhuro, V. (2021). Road images augmentation with synthetic traffic signs using neural networks. *Computer Optics*, 45(5), 736–748. <https://doi.org/10.18287/2412-6179-CO-859>
- Li, N., Song, F., Zhang, Y., Liang, P., & Cheng, E. (2022). Traffic context aware data augmentation for rare object detection in autonomous driving. In *2022 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 4548–4554). IEEE. <https://doi.org/10.1109/ICRA46639.2022.9811724>
- Luan, S., Li, C., Xu, P., Huang, Y., & Wang, X. (2023). MI-YOLO: More information based YOLO for insulator defect detection. *Journal of Electronic Imaging*, 32(4), Article 043014. <https://doi.org/10.1117/1.JEI.32.4.043014>

- Møgelmo, A., Trivedi, M. M., & Moeslund, T. B. (2012). Learning to detect traffic signs: Comparative evaluation of synthetic and real-world datasets. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR 2012)* (pp. 3452–3455). IEEE. <https://doi.org/10.1109/ICPR.2012.400>
- Qian, R., Liu, Q., Yue, Y., Coenen, F., & Zhang, B. (2016). Road surface traffic sign detection with hybrid region proposal and fast R-CNN. In *Proceedings of the 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)* (pp. 555–559). IEEE. <https://doi.org/10.1109/FSKD.2016.7603233>
- Raisa, J. F., & Yasmin, R. (2025). Bangla Sign Language Recognition: A Comprehensive Review of Machine Learning Approaches and Data Sources. *Journal of FST*, 3(1), 53–86. <https://doi.org/10.64494/JFST/jr/2025/03/53-86>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). 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). IEEE.
- Sekizawa, A., & Nakajima, K. (2019). Context-aware training image synthesis for traffic sign recognition. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications: Volume 5: VISAPP* (pp. 466–473). SCITEPRESS. <https://doi.org/10.5220/0007387904660473>
- Shakhuro, V., Faizov, B., & Konushin, A. (2019). Rare traffic sign recognition using synthetic training data. In *Proceedings of the 3rd International Conference on Video and Image Processing (ICVIP 2019)* (pp. 23–26). Association for Computing Machinery. <https://doi.org/10.1145/3376067.3376105>
- Shi, J., Rao, H., Jing, Q., Wen, Z., & Jia, G. (2024). FlexibleCP: A data augmentation strategy for traffic sign detection. *IET Image Processing*, 18(12), 3667–3680. <https://doi.org/10.1049/ipr2.13211>
- Shafi, M. M. S., & Mirza, J. F. (2025). An Explainable Ensemble Learning Framework for Brain Tumour Using Pretrained Models and XAI Techniques. *Journal of FST*, 3(1), 125–146. <https://doi.org/10.64494/JFST/v3i1/ms/2025/06/125-146>

- Tabelini, L., Berriel, R., Paixão, T. M., De Souza, A. F., Badue, C., Sebe, N., & Oliveira-Santos, T. (2022). Deep traffic sign detection and recognition without target domain real images. *Machine Vision and Applications*, 33(3), Article 50. <https://doi.org/10.1007/s00138-022-01302-0>
- Tabernik, D., & Skočaj, D. (2020). Deep learning for large-scale traffic-sign detection and recognition. *IEEE Transactions on Intelligent Transportation Systems*, 21(4), 1427–1440. <https://doi.org/10.1109/TITS.2019.2913588>
- Villalonga, G., Van de Weijer, J., & López, A. M. (2020). Recognizing new classes with synthetic data in the loop. *Sensors*, 20(3), Article 583. <https://doi.org/10.3390/s20030583>
- Wang, G., Jin, P., Qi, Z., & Li, X. (2025). Traffic sign detection method based on improved YOLOv8. *Scientific Reports*, 15(1), Article 19385. <https://doi.org/10.1038/s41598-025-03753-y>
- Wang, Y., Sun, W., & Liu, C. (2023). Cost-effective approach towards building a traffic sign data inventory using open street images [Technical report]. *Connected Cities for Smart Mobility Toward Accessible and Resilient Transportation*, University of Washington. <https://rosap.ntl.bts.gov/view/dot/72351>
- Yang, C., Zhuang, K., Chen, M., Ma, H., Han, X., Han, T., Guo, C., Han, H., Zhao, B., & Wang, Q. (2023). *Traffic sign interpretation in real road scene*. arXiv. <https://arxiv.org/abs/2311.10793>
- Zhao, L., Wei, Z., Li, Y., Jin, J., & Li, X. (2023). SEDG-YOLOv5: A lightweight traffic sign detection model based on knowledge distillation. *Electronics*, 12(2), Article 305. <https://doi.org/10.3390/electronics12020305>