Dijlah Journal of Engineering Science (DJES) Vol. 3, No. 2, June, 2026, pp. 308-320
ISSN: Printed: **3078-9656, Online: 3078-9664**, paper ID: 168

A Comparative Study of CNN, SSD and VGG16 for Robust Traffic Sign Recognition

Hasan Hammad Owaid[1]
[1]Ministry of education, Baghdad, Iraq

Article Info

Article history:

Received Feb., 26, 2026
Revised Mar.,27, 2026
Accepted Apr.,18, 2026

Keywords:

Traffic light,
Traffic jam
Traffic signs
Deep learning
Transfer learning

ABSTRACT

These techniques do everything in their power to get it right to avoid hazardous mistakes. The downside, they can also be too slow for everyday driving, where the decisions need to happen within a blink of an eye. Focus on Speed: Such methods run very fast. The downside, they may not be consistent enough at challenging times, such as reading signs far away, under darkness, or at bad weather. This thesis overcomes these challenges through combining deep learning-a sophisticated type of AI-with an intelligent image pre-processing technique. We have chosen deep learning because it has been proven to perform very well in a wide range of analysis task, even outperforming or matching human capabilities in certain visual tasks. In our tests, we implemented different deep learning models-CNN, SSD, and VGG16-based on the strengths and weaknesses of our image dataset. Our experience showed that the simplest CNN model was not fit for this purpose. On the other hand, the VGG16 model performed better. It was capable of recognizing traffic signs reasonably well, including under complex situations such as variable weather conditions or lighting, or when the traffic signs were far away. The best performing model is VGG16 with 91% testing accuracy.

Corresponding Author:

Hasan Hammad Owaid
Ministry of education
Baghdad, Iraq
Email: Hasan@gmail.com

1. INTRODUCTION (10 PT)

Just think of traffic signs as friends along the road that use pictures or symbols to represent information on directions, rules of the road, or warning drivers about something. You will find them mounted on poles beside or over the road. Their whole purpose is to keep you safe from accidents by preventing them and thoroughly informing you of what is ahead.Signs are there to perform specific functions: warning signs, like advance alarms, alert you to hazards you cannot see immediately, such as a sharp turn, people crossing, or farther down the road. They give you some time to react to whatever is ahead of you. Then, there are priority signs, like referees at an intersection, that decide who goes first in order not to collide. The "no" signs are restrictive signs, informing you about what not to do, such as "No Entry" or prohibition of certain types of vehicles. Mandatory signs, on the other hand, are signs that tell you what you "must" do: "Turn Left Only" or requiring minimum speeds.Apart from those, you also notice special regulation areas like quiet residential zones or school streets. Information signs are just useful bits of data like town names or where the next gas station is. And then there are direction signs, which are your navigators to show you the way to go for exits, turns, or your destination so you would not lose your way.Scientists have therefore been creating, for years, systems that can enable automobiles to automatically detect and interpret these signs, notably what is called Traffic Sign Recognition. Why? Sometimes drivers just cannot see signs. Maybe they are tired, distracted, or the sign is occluded or faded. That is critical for self-driving cars: they must find signs at the lightning speed and in precision with which to make their way safely in the current world.
These smart systems work two steps.

Then, detection: the car's camera reads the road for anything that is even remotely like a sign. This is tough, as signs may be covered by trees, buildings, or bad weather. If a possible sign is identified, the second part of the process-classification-kicks in. The system will use a sophisticated computer program to decide precisely what this sign is pointing to-a stop sign.? A speed limit sign? An indication of ice? Notwithstanding regulation by organizations such as the World Health Organization in matters of safe driving, accidents still happen. Most often, it is a question of two things: driver error (drivers absent or failing to notice signs) and bad signs (deteriorated, dirty, or hard to notice). It is for this reason that creating effective automatic traffic sign detection is important. It acts as a co-pilot, helping to pick up on what the human eye might not detect and to make important road information so ever easily comprehendible – making our roads safer for everyone. Others, including research [3], start with a "separate this from that" game. They are teaching computers to look at a photo and separate pixels into two sets: the ACTUAL sign (object) and the remainder (background). By comparing each infinitesimal speck of color to a reference point, the system can become master at picking signs hidden in ordinary color photographs.They, and others like authors in [4], divide it into stages. They initially train the system to learn colors for itself, followed by object edge positions by distance measurements. Finally, they interpret color patterns to determine what the sign is communicating.

Study [5] took a different route and cast colors on a 3D map (e.g., XYZ coordinates) in order to generate a virtual map. If the topology of a traffic sign wraps around this map in exactly the right way – bingo! The system recognizes it has seen a sign. Others, like those of [6], employed a "color memory bank." They learn the exact colors of each of different signs, and compare new images to what they perceive by this memory. Anything that's not close enough is skipped over by the system, then color double-checked using an analyzer.Concurrently, the authors of [7] developed shapes and symbols. They built modules that understand arrow positions and directions so that sign symbols can be interpreted by the computer as humans interpret them.To deal with ugly real-world images, authors of [8] and [9] used an electronic "eraser" (Laplacian filter) to clean images. They pre-filter visual noise, afterwards detect objects, and use an evolutionary algorithm (Genetic Algorithm) lastly to correctly identify signs. Bad weather conditions and poor visibility? That is what research study [10] solved head- classified shapes according to orientation to identify signs even when the visibility is poor.Researchers in [11] came up with an intelligent sorting facility (Hierarchical Spatial Feature Matching). They instructed the computer to learn where signs are typically located and ignore the others, thereby recognizing much faster. Experiment [12] combined two powerful approaches: they supplemented a neural network (CNN) with image processing and expert feature detectors (HOG). This was a master combination for decreasing false alarms but still being successful at detecting signs. For still better informed classification, the authors in [13] developed a carefull cost-sensitive neural network with great care. Their innovative 10-layer model with sampling and dropout attained accuracy of 86.5% – as evidence of taking great care.One of the feature success stories was from a research paper [7], in which researchers combined image processing with a high-level neural network (VGGNet). They used a standard traffic sign dataset and obtained close to perfect performance: 99% validation accuracy and 100% testing accuracy. Even better, they optimized the system by eliminating redundant data later. Here is a sentence to refer to this research: In their study, Al-Hitawi et al. [14] introduced a specialized toolkit designed to generate and augment datasets for Hungarian Handwritten Text Recognition such a techiques could be utlized to generate synthestic data.Real-time speed was the line [15] researchers took. They used color analysis (HSV model) to quickly detect sign candidates, cropped the areas and processed them through a neural network. When they experimented with different approaches using Bangladesh traffic signs, their neural network outperformed others at 97%. Study [17] also suggested a smart two-stage system with CNNs: sign shape initially, followed by recognizing its exact meaning. Though not suggested for use directly in real-time, it indicated how breaking down the problem is useful.Similar techniques can be used for enhanced the visualization of features in traffic sign recognition applications, as proved by Al-Hitawi et al.'s [16] evidence of the efficacy of transformer-based architectures in perceptual sequence mapping.The authors in [17] utilized highly accurate 99% with a thinner neural network (VGG16). They eliminated redundant layers and made use of intelligent processing techniques to realize optimal performance at reduced computing needs.Color plays a strong role in sign detection. All start by converting regular color photos to specialty format wherein color is separated from brightness allowing the computer to focus on the essentials. However, the accuracy is wildly inconsistent (35% to 99%) because of one gigantic problem: insufficient light. Countries with dark, long winters fare abysmally here. Shape-based methods fix color issues by decomposing images into basic black-and-white. But these struggle with other issues of the real world: signs far from the camera, or things that seem impossibly close to signs.

Neural networks are promising since they are taught on photographs directly, without directions. But most are too computationally intensive to work on real-time in moving cars.

1.1 Problem Statement

Traffic sign recognition (TSR) is an inevitable component of autonomous driving systems and intelligent transportation systems. Nevertheless, the detection and classification of traffic signs in real-world scenarios is not an easy task as there

may be several factors such as lighting conditions, occlusions, weather conditions, shape, color, and language differences in signs. These dynamic conditions have the effect of rendering traditional image processing techniques incapable of producing good results, and therefore more complex, robust, and adaptive recognition systems must be developed.

1.2 Aims of Research

To develop a robust traffic sign recognition model that can detect and classify different classes of traffic signs under difficult environmental conditions. To test how well traditional machine learning techniques and modern deep learning models perform regarding traffic sign recognition. To improve the accuracy, performance, and generalization ability of the TSR system by incorporating data augmentation and model tuning.

2. Dataset

2.1 . Data collection

The project adopts an experimental and quantitative method. The project is based on image acquisition and pre-processing of road signs, model development and training of various recognition models, and comparison of performances based on quantitative metrics. The method is model-based, experimented, and validated with real-world and benchmark data sets to ensure the reliability and scalability of the proposed system.

Figure 1: multiple signs.

However, this dataset isn't flawless; it presents us with some significant difficulties. Signs far down the road are captured in many pictures. These tiny indications result in a little "selection box" on the computer that obscures important information. You miss the fine print, just like when you try to read a street sign from a block away. This may reduce the accuracy of our model. Additionally, we have pictures that were taken at night or in poor light (see figure 2). Signs are blurred into the background in these dreary images, making it difficult for the computer to distinguish between them. This restricts our model's potential performance in dimly lit environments. How can we address these constraints, then? We use our imagination for the dark images.

Figure 2: photos taken in low light.

We will perform data augmentation. Consider that we are providing our model with additional practice by producing variations of the existing photographs by applying various lighting effects and illuminating them.
This will condition our model to be capable of recognizing signs even when it's dark. For the distant signs, digitally zooming in on the important signs in those pictures can help our model "see" them better, compensating for the tiny original size. And for the dense scenes with a lot of signs, we'll particularly train and design our model to be a multi-object detection specialist detecting and identifying numerous signs in a dense image. Solving this head-on will make it much stronger.Finally, this GTSDB dataset is both a blessing and a curse. Its realism and variety are great for building an efficient model. But distant signs and low light issues are real weaknesses we cannot ignore. Being conscious of these advantages and disadvantages beforehand, we can choose the right tools and design our model accordingly to overcome the challenges. This pre-exposure is critical to the establishment of a system that not just functions in the lab, but operates reliably out on the actual road.

2.2 . Data preprocessing

The previous dataset description can be summarized as in Figure 3. The preprocessing-based database pons and cons steps proposed by this paper are discussed in the following subsections in the methodology.

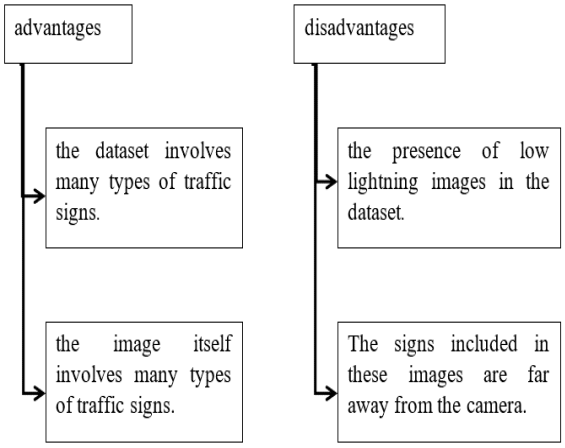

Figure 3. Data description.

3. METHOD

This study uses a quantitative and experimental approach. Road sign picture capture and pre-processing, model creation and training, and performance comparison using quantitative measurements are all included in the project. To guarantee the dependability and scalability of the proposed system, the approach is model-based, tested, and assessed using benchmark and real-world data sets.Data Collection and Preprocessing: Compiling sets of traffic sign data and carrying out preprocessing operations such color correction, noise reduction, normalization, and resizing. Feature Extraction: Automatic feature extraction using computer vision-based Convolutional Neural Networks (CNNs) and conventional techniques (HOG, SIFT). Model training involves the use of CNN-based models (e.g., LeNet, AlexNet, or ResNet) and classification methods like Support Vector Machines (SVM) and Random Forest. Model performance is evaluated using metrics such as accuracy, precision, recall, F1-score, and confusion matrices. Optimization is the process of maximizing performance and preventing overfitting through regularization, data augmentation, and hyperparameter adjustment.

3.1 . Data preprocessing

Image resizing constitutes a fundamental preprocessing step in preparing datasets for traffic sign recognition models. This standardization involves uniformly scaling all input images to a fixed dimension (e.g., 224x224 pixels in this work), yielding significant advantages. Firstly, by focusing the model's learning ability on the important aspects of the traffic signs themselves, resizing lessens the influence of extraneous background data. Second, it deals with the intrinsic variation in image sizes caused by different distances and capture equipment (phones, cameras), which would otherwise make feature classification and model training ineffective. Thirdly, standardizing image dimensions significantly lowers training time and computational load. This is particularly important for computationally expensive models such as CNNs, where processing effort grows with image size. Finally, by training on consistently formatted datasets, scaling enhances model generalizability and performance on unseen real-

world data. One of the major methodological issues when resizing is preserving the dimensions of bounding boxes and their relative locations, if any, to maintain exact localization information in the reshaped image space.
The Python Programming is used to access image and label datasets for navigation through directories in the preprocessing pipeline used in this work. Subsequently, picture scaling operations have been performed by the PIL library, standardizing all input images at the same size of 224 x 224 pixels. After resizing, the modified images were saved into the target images directory. Importantly, matching label files have been read while resizing the image to update metadata. In particular, bounding box coordinates have been updated to maintain spatial accuracy related to enlarged photos. Notably, the axletree. Element Tree library facilitated label synchronization, crucial for maintaining object localization integrity by allowing parsing and updates of XML-based label files.

3.2. Normilzation and Standarization

Color normalization is the second important preprocessing stage of our pipeline in traffic sign classification. Since the semantical meaning of traffic signals, like the red octagon for stop, intrinsically comes from color, variations due to lighting conditions peaks per channel; deviations such as skewness are also measured. Secondly, color range analysis, which finds minimum and maximum pixel intensities for each channel, can detect outlier values or values with extreme ranges that may highly influence normalization.Normalization primarily assists the model in learning the intrinsic color semantics required for traffic sign interpretation, besides reducing computational inefficiencies associated with learning varied color representations at times of varying sunlight intensities. Normalizing the intensity range creates a consistent feature space and enhances the model's ability to generalize to previously unknown data taken under various lighting scenarios. In the implementation, we will utilize the OpenCV library, cv2. We split each image into its RGB channels. Channel-wise Min-Max Normalization is applied to scale intensities to [0,1]. This is done by subtracting the channel's minimum intensity value and then dividing by its range max - min. The standardized image is then constructed by combining these normalized channels back together. Conditions introduce considerable variation in color representation across the dataset. Normalization methods are, therefore, necessary to reduce these variations and make the model robust. The most common techniques include histogram equalization that balances the histogram through redistribution of the intensities of the pixels, and intensity scaling, which normalizes into a common range, usually within the range of [0, 1]. Such selection of the best normalization method requires close examination of the color distribution of the dataset. In particular, in this work, computation of the mean (average intensity) and standard deviation (std, which is a measure of dispersion, telling something about how spread out the intensity values are) of each of the RGB channels across all photos is done. While standard deviation measures the amount of color variation, the mean shows possible color bias. To examine the shape of the distribution within the default 8-bit range [0,255], histogram analysis is also conducted for each channel. An ideal distribution would be characterized by smoothness and recognizable peaks per channel; skewness and any other deviations are considered. Additionally, color range analysis, which identifies the minimum and maximum pixel intensity for each channel, finds outliers or extreme values that can have a disproportionately large effect on normalization. Normalization mainly helps the model learn the intrinsic color semantics necessary for traffic sign interpretation, apart from reducing the computational inefficiencies related to learning varied color representations under varying sunlight intensities (direct vs. indirect). Standardizing the intensity range creates a consistent feature space that increases the model's ability to generalize on previously unseen data recorded across different lighting conditions. We use the OpenCV (CV2) library in implementing this step. Every image is decomposed into its different channels of RGB. Channel-wise, min-max normalization is performed by subtracting the minimum intensity value for each channel and dividing by its range (max - min) to scale intensities to [0, 1]. The standardized image is reconstructed by re-combination of the normalized channels.

3.3. Denoising

The third key preprocessing step in our traffic sign recognition pipeline is noise reduction. Image noise can mask the discriminative information that is needed for accurate object recognition and may result from compression methods or sensor anomalies. This degradation conceals distinctive patterns and features, hence directly impacting model performance. This further makes feature extraction approaches more error-prone, for example, misinterpreting noise-induced edges as the boundary of signs. Besides, too much noise lowers the efficiency in processing by putting a heavy computational burden on the system. We use three quantitative metrics: MSE, PSNR, and STD to select the best approach for denoising.Lower values indicate better noise suppression. MSE measures the average squared difference between pixel values in the original noisy image and a reference noise-free image. Higher values indicate better reconstruction quality relative to the original image. PSNR, expressed in decibels (dB), is the ratio of the maximum possible signal power to the corrupting noise power. Although distinguishing between noise reduction and the suppression of valid image contrast is important, a significant drop in STD after denoising indicates lower random intensity variations and higher uniformity. STD quantifies the dispersion of pixel intensities. Denoising

approaches can be roughly categorized as non-linear or linear, such as convolution-based filters. In the present review, we evaluate the following six denoising techniques:

- Gaussian filter.
- Median filter.
- Bilateral filter.
- Brightness Gaussian filter.
- Non-Local Means filter.
- Impulse Removal filter.

3.4. Data augmentation

Data augmentation is the first preprocessing step proposed in this thesis. In this approach, the actual size of a training dataset is artificially increased by applying controlled transformations-flipping, rotation, or scaling-on preexisting samples. Augmentation is a well-known deep learning technique that overcomes some of the disadvantages of creating huge and heterogeneous real-world datasets, such as in TSR. It exposes a wider variety of feature representations to the model without requiring the acquisition of extra data by producing synthetic versions. There are three reasons why TSR should be increased. Initially, it strengthens the model's resilience to actual variances found in traffic sign imagery, such as alterations in camera perspective (such as angular displacement), lighting, and (such as variations brought on by the weather), and partial opacity. This includes enriching the feature space during training, allowing for the uncovering of latent properties not well represented in the original dataset. Thirdly, it solves the problem of class imbalance, in which some categories of signs, such as "Speed Limits", are overrepresented in comparison with others like "Stop" signals. Augmentation decreases model bias toward dominant categories by artificially increasing samples for underrepresented classes.

4. MODEL TRAINING

In this section, three different TSR architectures are presented: CNNs, SSDs, and VGG16 with transfer learning utilization. Each of them has distinct advantages when dealing with TSR difficulties.

Convolutional Neural Networks (CNNs) are naturally good at extracting spatial features directly from pixel information, making them an obvious choice for image-based tasks. In the case of TSR, where recognition relies on spatial correlations between shapes, colors, edges, and color blobs-for instance, the typical stop signs octagonal shape and red-white coloration-this capability is inherent. A CNN learns to build up basic features into complex representations-through a set of hierarchical convolutional layers-suitable for class discrimination. The automatic feature extraction during training, which eliminates the manual development of spatial feature descriptors, is a considerable advantage. Single Shoot Detection (SSD), designed for real-time applications like TSR, SSD technology is a single-stage object recognition architecture that offers significantly faster speeds compared to traditional two-stage detectors. Its key advantages include:

- Unified detection: Detects and classifies objects in a single feedforward pass.
- Adaptive selection boxes: Utilize the concept of virtual bins for multiple feature map sizes, then learn the variation during training to accurately match signal dimensions. This provides greater versatility than simple CNN systems.
- Multi-scale feature merging: This method enhances detail and signal detection by merging hierarchical features of varying convolutional depths using FPN.
- Interference handling: This method uses class confidence scores for each predicted box, allowing for efficient masking of non-maximal values to eliminate overlapping detections, which is essential in multi-marker scenes.

5. RESULTS AND DISCUSSION

5.1. Prepossessing Results

In Chapter 3 describes the data preprocessing technique proposed in the current study to account for the differences in the data set. We proceed with the current section by reporting the results regarding the main objectives one and two. A complete statistical analysis of the value of the intensity of the different channels is necessary to understand the type of differences that exist in the data set. We will focus on the mean, standard deviation (std), minimum (min), maximum (max), and quartile measurements. The result will be summarized in the following table. The blue channel has an extremely high mean value compared to the other channels, meaning that the mean value of B_mean was 144.27261, which is the central point around which the majority of the pixels are concentrated. Such a high value leads to the skewed representation of the visual data with respect to the blue color.Turning to the standard deviation, the problem is serious. The numbers indicate that the std values are high across all the channels. For example, the blue, green, and red channels have standard deviations of about 36, 31, and 28, respectively. The high

numbers demonstrate variability in the entire data set. The variability levels are so high that they can be regarded as noise, which may hide the crucial information required by the model to learn during the learning process. Finally, analyzing the minimum and maximum values indicates that the channels support a wide band of spectrum. For instance, the average value of B_min is 30.0992. Also, the average value of B_max reaches 248.9681. Moreover, the 75th percentile of B_min is 46, while the 75th percentile of B_max is 255.

Table 1: RGB Mean Values.

	B_mean	G_mean	R_mean
Mean	142.8569	135.9324	118.6745
Std	35.9213	32.1846	28.1479
Min	15.4382	11.2053	5.6847
25%	120.4835	117.8632	101.9248
50%	144.2951	139.6721	118.2037
75%	165.9127	157.3048	135.4162
Max	228.3419	232.5086	235.6721

Table 2: Min RGB.

	B_min	G_min	R_min
Mean	31.5847	27.4821	16.3928
Std	22.1389	18.9576	17.2354
Min	0	0	0
25%	11.2043	8.1746	1.2389
50%	33.6951	29.1027	12.5617
75%	48.3278	43.6505	29.4782
Max	97.4182	101.2254	132.9471

One final area of inquiry remains: a review of the histogram Figure 4.

Table 3: Max RGB.

	B_max	G_max	R_max
Mean	247.6523	249.7138	248.9825
Std	13.0286	11.8752	12.5461
Min	90	84	110
25%	246	252	250
50%	254	255	255
75%	255	255	255
Max	255	255	255

The next figure 4 shows the histogram for the blue channel.

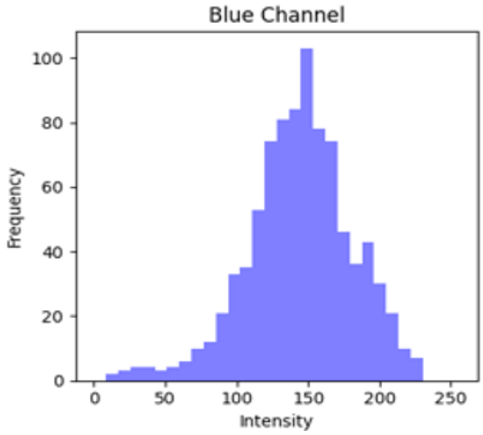

Figure 4: The histogram of the blue channel.

The drawing shown above is the graphical representation of the feature centered on the average value magnitude. The most recent result available had its result common with the previous trend, hence verifying the ubiquitous presence of blue levels of colors in the image components. Moreover, the drawing also portrays the distribution of image points with some having highest/lowest scores, hence protruding towards the extremes.The salient feature of the histogram related to the green components is presented in the succeeding figure, Figure 5. The result shown below in Figure 4 is analogous to the result presented in Figure 4, but with a different spread pattern. The result for the red component is presented below in Figure 6.

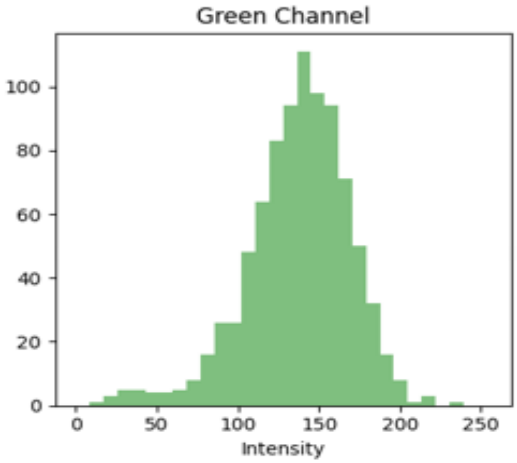

Figure 5: Histogram of the green channels:

The resultant image is characterized by well-defined shapes, while the geometric shapes involved have less importance than the blue and green channels' colors. The current output is characterized by less intense crimson compared to the blue and green channels' colors. However, taking into consideration the high standard deviations indicated by the datasets presented in the previous screenshots, the application of the normalization process is highly advocated for the data analysis process.

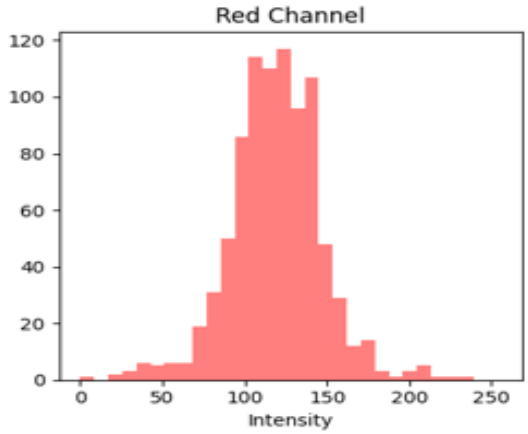

Figure 6: The histogram of the red channel.

A mention is made about section 2, step 3, in which the three prominent parameters Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Standard Division, i.e., Standard Deviation (STD given for the assessment of the quality of the image are described. The result values are available in table 4.

Table 4: A comparison of the different techniques used to remove noise.

	Average MSE	Average PSNR	Average STD
Gaussian Blur	34.27	27.11	57.19
Median Filter	31.46	25.18	58.25
Bilateral Filter	28.71	32.38	59.10
Impulse Removal	13.28	36.36	59.99
Gaussian Filter	34.27	27.11	57.19
Non-Local Means	30.03	30.79	59.65

Looking at the results found on the table above; it can be deduced that the impulse suppression method has the smallest MSE amounting to only 13.28 among the six noise removal techniques that were analyzed. Thus, this technique gives the best result in terms of the size of the pixel values. In addition, the impulse suppression technique also has the highest PSNR value at 36.36, far superior to all the others. So, it would then follow that the impulse removal method is singularly effective in producing the optimal image quality as compared with all the other methodologies under test. The second observation is that all of the algorithms employed have a relatively high STD that falls between a narrow range of 57 to 59.

5.2. CNN results

Equations Shown in Figure 7 are the results obtained using the CNN process on the TSR project, along with a sidebar of some interesting information. A few problems are quite obvious right off the bat; of these, the first problem is that of the ill-positioned bounding boxes. This problem generally results from the limited number of image samples present in the training dataset. Because of the limited representation of traffic signs within various operating scenarios, the performance of the model will be much less than optimal when tested on data it has never seen before.

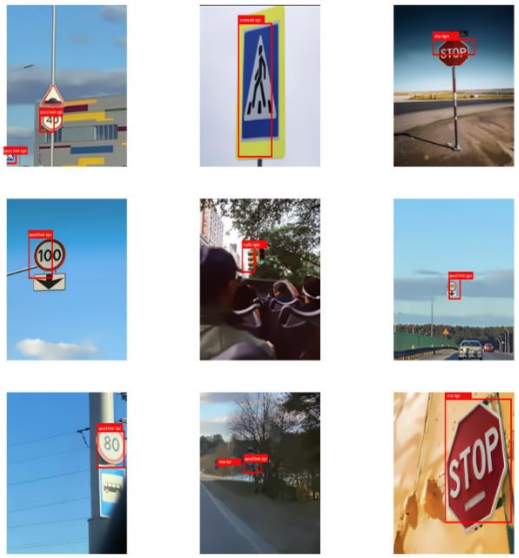

Figure 7: CNN model results.

The next challenge concerns the scale variation. The dataset used for training consists of a mostly predetermined size of traffic signs, so, when exposed to new, unseen data, the model tends to incorrectly estimate the size, either larger or smaller, due to a lack of preparation to distinguish different-sized traffic signs. This problem, discussed earlier, can be due to the lack of relevant training samples. Another possible situation where this may occur is when one or a few types of traffic signs dominate in number over others. In other words, if the population of one type of sign class is much larger, the model will always fall back to uniformly scaling the traffic signs based on specifications from this overly represented class. The third limitation is related to the restricted capabilities regarding the detection of multiple objects. As one can notice in Figure 7, the CNN can detect a maximum of two objects with good accuracy. This means that in the case of overlapping signs, the effectiveness of this model decreases a lot, since it: That is, it will either group all superimposed signs into a single bounding box, or worse, not recognize some signs altogether.

5.3. SSD results

Results obtained by the Single Shot Detector model, represented schematically in Figure 8, will be discussed in detail within this section. As is evident from the relative performance comparison, the SSD model significantly outperformed the CNN architecture for issues involving bounding box localization.

Figure 8: SSD model results.

In the multi-object case considered, the maximum predicted value was four symbols. This assertion is supported by the significant overlap witnessed between the objects. Moreover, a large number of overlapping objects pose a crucial challenge in addressing this problem at hand. Moreover, it appears that the model was not expecting such an outcome. This challenge is parallel to what happened with the CNN model; it is due to scale differences. Although the SSD model did an excellent job implementing the FPN framework and exploiting its benefits, the gap observed here indicates that there is substantial heterogeneity in the scale of the symbols in the dataset. This eventually leads to a decrease in the FPN performance in collecting valuable features.

5.4. VGG Results

Figuer 9 shows the forecast outcome of the identification process using VGG16, proving that it is superior because it performed far better than the CNN and SSD models. The bounding boxes perfectly capture the actual measurements of the road signs. Similarly, the model identified multiple items within the frame and proved the potential of its operation.

Figure 9: VGG16 model results.

These subsequent results are evidence that the present model has the ability to capture minute features effectively, which may be related to boundary handling, contours, and numerous other aspects. After this, the model's performance will be fine-tuned to understand "aa" symbols present in the image and to provide enhanced discrimination among them. Further, the crucial role of VGG16 deep network architecture cannot be ignored, as this allows the effective integration of complex variability associated with the said features, thereby converting these data into a strong paradigm for discrimination among different traffic signs.

6. CONCLUSION

This research has been motivated by the need to address the TSR problem because of its wide applicability in different scenarios. This is a preliminary academic exercise that attempts to deconstruct the relevant dataset in terms of its quantitative measurements and innate limitations. Our analysis reflects that the dataset is an asset that is contradictory to some degree, as much of the variety and numerousness of the images form a core ingredient for model performance.Limitations in the current dataset being analyzed seem to manifest most through two dimensions. One issue is that there is a lot of variation in distances between the photographic device and the object taken. The other major barrier to optimal accuracy in this case is the strong fluctuation in ambient illumination conditions.Drawing from the intuition obtained in the dataset analysis, we propose a four-step data preparation procedure. First of all, in order to improve the generalization capability of the model, all images were resized uniformly to a fixed dimension of 224x224 pixels. Then, we analyze the chromatic characteristics of the images with a group of four statistical measures to determine the best normalization method. After that, the performance of six different methods for noise removal is evaluated using MSE, PSNR, and STD as three most important metrics for performance evaluation.The impulse removal technique presented better results as per MSE and PSNR in the findings from this research. This particular technique minimized noise while being able to preserve key features, thereby making the resultant noise-free image faithful to its original counterpart.Concerning the classification goal, the results are provided in Table 5. The values from Table 5 convincingly indicate that the CNN was the weakest in handling the TSR task and did not reach the target of 88% accuracy. However, the SSD was remarkable in bounding box accuracy and also on multiple detections relative to the CNN. Therefore, VGG16 was the most promising model candidate for handling the complexities involved in TSR.

Table 5: Accuracy results summary.

	CNN	SSD	VGG16
Training	68%	71%	**89%**
Testing	89%	90%	**91%**

.

REFERENCES

[1] M. P. Philipsen, A. Møgelmose, T. B. Moeslund and M. M. Trivedi, "*Vision for Looking at Traffic Lights: Issues, Survey, and Perspectives,", IEEE transactions on intelligent transportation systems*, 2016, 17.7, 1800-1815.

[2] K. He, X. Zhang, S. Ren and J. Sun, "*Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in Proceedings of the IEEE nternational conference on computer vision*, pp. 1026–1034, 2015.

[3] G. D, L. S and Y. X, "*Recognition of traffic signs by artificial neurale networks*," *In Proceedings of ICNN'95-International Conference on Neural Networks*, 1995, Vol. 3, pp. 1444-1449.

[4] K. N and E. L, "*A real-time histography approach to road sign recognition," In Proceeding of Southwest Symposium on Image Analysis and Interpretation*, 1996, pp. 95-100.

[5] Y. N, A.-A.-N. D and M. M, "*Road traffic sign detection in color images," presented at 10th IEEE Inter. Conf. on Electronics, Circuits and Systems (ICECS 2003)*, Sharjah, United Arab Emirates, 2003, Vol. 2, pp. 890-893.

[6] C. F. S. C. a. P. Y. C. Fang, *"A road sign recognition system based on dynamic visual model," In 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings*, 2003, Vol. 1, pp. I-I.

[7] R. L. a. V. R. L. Priese, *"Ideogram identification in a realtime traffic sign recognition system," In Proceedings of the Intelligent Vehicles'* 95, 1995, pp. 310-314.

[8] Y. A. a. T. Asakura, *"A study on traffic sign recognition in scene image using genetic algorithms and neural networks," In Proceedings of the 1996 IEEE IECON. 22nd International Conference on Industrial Electronics, Control, and Instrumentation*, 1996, Vol. 3, pp. 1838-1843.

[9] Y. A. a. T. Asakura, *"Detection and recognition of traffic sign in scene image using genetic algorithms and neural network," In Proceedings of the 35th SICE Annual Conference. International Session Papers*, 1996, pp. 1343-1348.

[10] C. Schiekel, "*A fast traffic sign recognition algorithm for gray value images," In International Conference on Computer Analysis of Images and Patterns, Berlin, Heidelberg: Springer Berlin Heidelberg* 1999, pp. 588-595.

[11] P. P. a. J. Novovicova, *"Road sign classification without color information," In Proc. of the 6th Annual Conference of the Advanced School for Computing and Imaging*, 2000.

[12] L. Shangzheng, *"A Traffic Sign Image Recognition and Classification Approach Based on Convolutional Neural Network," in 2019 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, 2019, pp. 408–411.

[13] T. S. Tsoi and C. Wheelus, *"Traffic Signal Classification with Cost-Sensitive Deep Learning Models," in 2020 IEEE International Conference on Knowledge Graph (ICKG)*, Aug 2020, pp. 586–592.

[14] M. A. S. Al-Hitawi, A. L. Alzaidy, M. A. Alazaizi, and M. A. Tharthar, *"Toolkit for Generating and Augmenting Hungarian Handwritten Text Recognition Dataset," Dijlah J. Eng. Sci*., vol. 3, no. 1, 2026.

[15] P. Dhar, M. Z. Abedin and T. Biswas, *"Traffic sign detection — A new approach and recognition using convolution neural network," in 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, 2017, pp. 416–419.

[16] Al-Hitawi MAS and Máté GN. Enhancing Transformer-Based Language Models for Hungarian Handwritten Text Recognition [*version 1;peer review:1 approved with reservations]. F1000Research 2026, 15:181* (https://doi.org/10.12688/f1000research.176408.1).

[17] Y. Wu, Y. Liu, J. Li, H. Liu and X. Hu, *"Traffic sign detection based on convolutional neural networks," in 2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*, 2019, pp. 1–6.

BIOGRAPHIES OF AUTHORS

Lecturer **Hasan Hammad Owaid** is a lecturer at the Ministry of Education, Baghdad, Iraq. He is currently engaged in academic and educational activities within the Iraqi education sector. He can be contacted through the Ministry of Education, Baghdad, Iraq.