Utilizing the Deep Learning model for Traffic Sign Recognition in Arbaeen Pilgrimage

Lect. Dr. Safa S. Abdul-Jabbar
University of Baghdad / College of Science for Women / Department of Computer Science
safa.s@csw.uobaghdad.edu.iq

Asmaa Ali Mazhar
Asma.Jabr1901@csw.uobaghdad.edu.iq

Abstract:

The Arbaeen pilgrimage is one of the largest human gatherings in the world, posing significant challenges for traffic management and traffic sign supervision due to extreme population density, frequent changes in temporary traffic routes, and the physical and mental fatigue experienced by organizational staff. These factors make it increasingly difficult to guide visitors, particularly the elderly and those unfamiliar with the area. This research aims to address these challenges by developing an intelligent, automated system capable of instantly recognizing Arabic traffic signs and providing audio alerts to drivers and traffic coordinators. A convolutional neural network (CNN) model was employed, incorporating data analysis techniques such as preprocessing, data augmentation (including rotation and zooming), and normalization. The system was trained and tested on a publicly available dataset (ArTS Dataset) containing Arabic traffic signs under diverse conditions (e.g., lighting, angle, clarity). An interactive graphical user interface (GUI) was designed, allowing users to upload images of traffic signs (simulating camera input) and receive real-time predictions accompanied by audio pronunciation in Arabic. The model achieved test accuracy exceeding 90% when evaluated on the designated test dataset, along with strong performance in additional metrics such as precision, recall, and F1-score. Although the data used does not originate from actual traffic signs in Karbala during the Arbaeen pilgrimage, it reflects similar real-world conditions. This work represents an initial step toward future field deployment, particularly in smart transportation systems, where such a system could support visitors in their native language and in real time, thereby enhancing public safety and mobility during large-scale religious events.

Keywords: Traffic Sign Recognition, Arbaeen Pilgrimage management, Traffic Sign Detection, Traffic management, CNN-based Model.

Introduction

One of the most important cornerstones of intelligent transportation systems is providing self-driving vehicles and advanced driver assistance technologies that can recognize traffic signs, which helps provide safe and easy driving. These systems must also provide a reliable approach to instantly identifying and translating traffic signs to ensure safety while driving, and support compliance with traffic regulations. Although there is clear progress in the field of traffic sign recognition systems, these systems still face many challenges that hinder their implementation and development as required (Alamri, S., & Kanwal,2024). Overcoming these challenges and providing high accuracy when adopting these systems in classifying traffic signs is an essential step before starting to implement any proposed smart transportation system. The most important challenges that can be addressed in these systems are the accuracy and quality of images, different lighting, as well as the difference in the design of traffic signs between countries. It is worth noting that these difficulties increase significantly when dealing with traffic signs in Arab countries, because these countries suffer from a lack of adherence to unified international standards when designing these signs, and they also contain special signs and patterns that distinguish them. Therefore, the process of identifying them is considered more complex and constitutes a real obstacle to Building effective TSR systems capable of adapting to a variety of operating environments (Farzipour, A.,Manzari,O.N.,&Shokouhi, n.d.). Several technologies have emerged that have promising capabilities and are able to address many of the challenges related to processing digital images efficiently,quickly, and with high accuracy. These technologies rely primarily on artificial intelligence (such as deep learning, especially convolutional neural networks (CNNs)). These techniques are distinguished by their effective ability in classification tasks, which makes them able to identify the most important sequential and graded characteristics of images. However, CNN relies heavily on the availability of large, properly classified datasets that are used to train these networks. Fully connected layers,which are damaged, may suffer from poor generalization and may not be as efficient as required in real-time applications (Bhuvaneshwari, B., Abbijeet, R., Singh, S., & Priyadharshini, 2025) (William, M. M., Zaki, P. S., Soliman, B. K., Alexsan, K. G., Mansour, M., El-Moursy, M., & Khalil, 2019). In three popular techniques for traffic sign identification are currently being utilized. practiced: machine,form (Wali, S. B., Hannan, M. A.,Hussain,A.,&Samad, 2015), and color-based (Ritter, W., Stein, F.,& Janssen, 1995),methods based on learning (Fleyeh, 2004). In order to provide a powerful and scalable Arabic TSR system specifically designed to help improve sign classification accurately and efficiently, this research includes a graphical user interface (GUI) that provides users with a clear and easy-to-use interface. The proposed system also provides the feature of audio output in Arabic describing traffic signs to the driver, thus enhancing accessibility for Arabic-speaking users and facilitating and improving the practical application of the system in real-world environments. This research explores the integration of alternative classifiers into CNN that aim to improve accuracy while maintaining computational efficiency.

Related works

TSR has become an important subject, leading many researchers in the field of Artificial Intelligence to propose many approaches for enhancing accuracy, efficiency, and processing time especially with the emergence of machine learning (ML) and deep learning (DL). In this section a number of recent papers will be reviewed which have significantly contributed to the growth of TSR technologies. In 2021, research was presented aimed at developing a real-time TSR system using artificial intelligence and deep learning techniques, specifically the YOLOv4 model of bypass neural networks (CNN).The proposed system also provides the feature of informing the driver via voice of the sign content, which helps reduce accidents resulting from Ignoring or not understanding traffic signs. Several versions of YOLO have been adopted (including YOLOv3 and YOLOv4-tiny). The best performance was (YOLOv4-tiny), which achieved an average accuracy (mAP) of 64.71% and a detection speed of 55 frames per second (FPS), which is suitable for real-time applications (Manawadu, M., & Wijenayake, 2024).

While in 2023, a lightweight model of bypass neural networks (CNN) for traffic sign recognition was developed, designed to achieve high accuracy and fast response time, making it suitable for ITS applications, especially in vehicles with limited resources. The model was trained on the GTSRB and BelgiumTS datasets, and achieved robust results using a few coefficients. He also adopted optimization techniques such as magnification, normalization, and smoothing to improve generalization and reduce bias, along with the use of Adam's algorithm and the cross-entropy loss function.

Comparison with well-known models such as GoogleNet,AlexNet, VGG, ResNet, and MobileNetv2 has shown the superiority of the model in terms of accuracy, performance efficiency, and low resource consumption (Khan, M. A., Park, H., & Chae, 2023). While In 2024 a real-time traffic sign recognition (TSR) system was developed,supported by voice assistance. The system is based on a convolutional neural network (CNN) trained on traffic light data, where the sign is recognized and then voiced to alert the driver. Pre-processing and performance optimization techniques using the YOLO model were employed to achieve a detection speed of up to 55 frames per second (FPS) and an average accuracy of 64.71%. The system aims to improve traffic safety,especially in environments where traffic signs are not clearly visible or familiar to drivers, while providing a low-cost system that can be used in autonomous vehicles or as a driver aid (kolluri,o.,Veldandi,S.,& Pittala,2024). More recently, in 2025, a cost-effective and scalable system for real-time self-driving applications was proposed. By combining advanced deep learning techniques and embedded systems by taking advantage of deep convolutional neural networks, the system achieves high accuracy and robustness in classifying traffic signs. This system is a practical approach to applying advanced driver assistance features on low-cost platforms, which enhances the field of vehicle technology (N G, G. K., Kishore, A., & Krishna, 2025) Also, in the same year, research was presented by Benfaress et. al. on an advanced traffic sign recognition system using a deep CNN network supported by explanatory intelligence (XAI) technology through the Grad-CAM tool. This helps the system interpret its decisions visually and helps address challenges that may be encountered. It is encountered by the driver such as poor lighting, fog, low-resolution images, and partially obscured signs, which affect recognition accuracy.

In this system, a high accuracy of 99.62% was achieved without data enhancement and 99.06% when using enhancement techniques due to the XAI technology that was used. (Benfaress, I., Bouhoute, A., & Zinedine, 2025). To simplify the presentation of the previously published research, the key studies discussed above are summarized in the following table.

Table (1): Related Works Summary

Research Year	Authors	Techniques Used	Results	Key Findings
2021	Manawadu, and Wijenayake	YOLOv3, YOLOv4, YOLOv4-Tiny, CNN, TTS (gTTS), CUDA, Open CV	55 FPS, 64.71% mAP, Precision: 0.79, F1 Score: 0.69	Voice-assisted system improves driver awareness; YOLOv4-Tiny best for real-time; suitable for embedded systems in vehicles.
2023	Khan et al.	Lightweight CNN, ReLU, Global Average Pooling, Adam Optimizer, Data Augmentation, Grid Search	GTSRB: 98.41%, BelgiumTS: 92.06%, Time: ~1264 sec, Params: 2.61M	Proposed model outperforms state-of-the-art models with fewer parameters; ideal for embedded systems in real-time TSRs.
2024	Kolluri et. al.	CNN (YOLO-based), Image Data Generator, Voice Assistance	Detection speed: 55 FPSAccuracy: 64.71% average	Developed a real-time voice-assisted TSR system with CNN, demonstrated viability for low-cost deployment.

Research Year	Authors	Techniques Used	Results	Key Findings
2025	Kishore et. al.	CNNs, YOLO-based TSR pipeline, data augmentation, grayscale conversion, Arduino-based control system, ultrasonic and IR sensors, sensor fusion	- Accuracy: 99.63% (GTSRB), 99.68% (CTSD)- Latency: 24–26 ms (with signs), 24–30 ms (without signs)-	Developed a real-time TSR system integrated with autonomous vehicle control. Combined high-accuracy CNN classification with low-cost Arduino-based obstacle avoidance and line following using sensor fusion. Offers a scalable, real-time solution for autonomous navigation.
2025	Benfaress et. al.	Custom CNN + Grad-CAM + Data Augmentation (DA) + XAI + Comparison with GoogLeNet, ResNet, etc.	Accuracy: 99.62% (no DA), 99.06% (with DA); F1-score: 99.67%	Robust recognition under adverse conditions; explainable decisions using Grad-CAM; outperforms state-of-the-art models.

Methodology

This section presents the methodology adopted in the design and development of the Arabic Traffic Sign Recognition System (TSR), which aims to classify traffic signs automatically and in real-time, which enhances Traffic Safety and supports intelligent assistance systems for drivers,especially in Arabic countries. The proposed model, as shown in Figure 1,is based on a set of interrelated steps, starting with the preparation of data, passing through the design and training of the model, ending with the implementation of an interactive graphical interface with voice feedback. The proposed model was implemented using the Python language, relying on the TensorFlow and Keras libraries to design the bypass neural network (CNN), and the Tkinter interface to develop the graphical interface, with the integration of the Pygame library to generate audio outputs in Arabic.

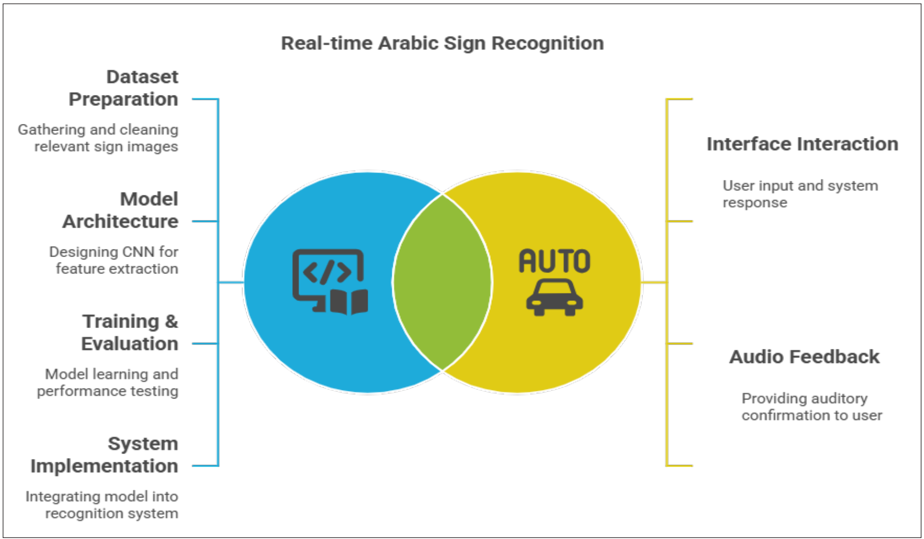


Figure 1: The main process of the proposed model

Each process consists of several steps, including dataset preparation, data processing funnel, CNN feature extraction, CNN data transformation, model training, and feedback, as illustrated in Figure 2.

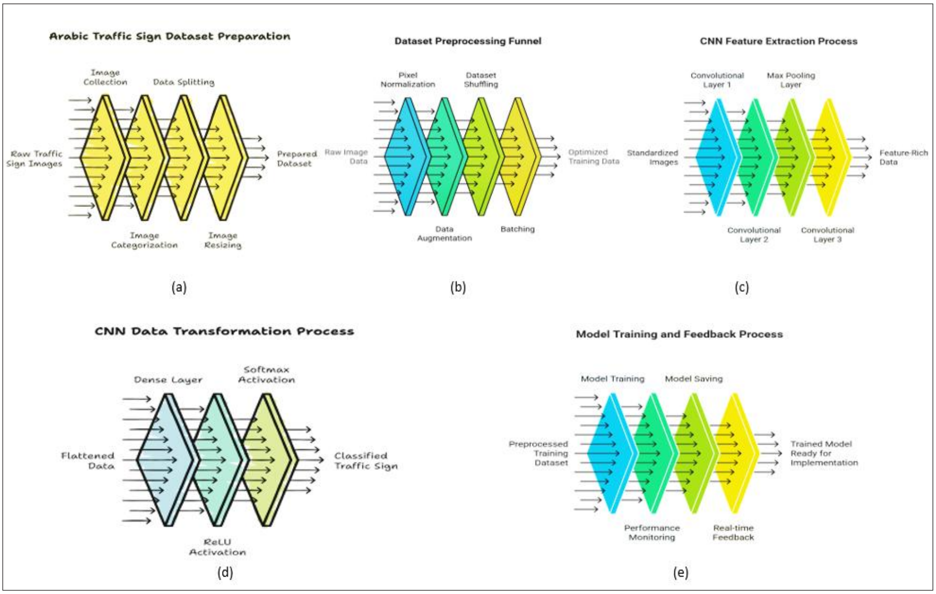


Figure 2: The internal steps of each process in the proposed model

Dataset properties:

The ArTS (Arabic Traffic Sign) dataset was adopted in this research and is one of the first specialized collections focused on Arabic traffic signs. It was officially published in March 2020. The dataset includes 24 categories of the most common Arabic traffic signs, with real-world images collected from three interconnected cities in the Eastern Province of Saudi Arabia: Al-Khobar, Dammam, and Dhahran. The dataset consists of:

- 2,718 real field-captured images, randomly divided into 80% for training (2,200 images) and 20% for testing (518 images).
- 57,078 augmented images, including 46,200 for training and 10,878 for testing, provided in compressed files due to their large size.
- This dataset offers diversity in lighting conditions, camera angles, and sign clarity, making it well-suited for training robust Arabic traffic sign recognition models in realistic environments.

Data Preparation Process:

The data preparation included the collection of images of Arabic traffic signs from open sources as well as data collected manually to cover variations in lighting, angles, and partial obscuration. The data was divided into:

- 70% for training
- 30% for testing

All images were converted to uniform dimensions (180×180 pixels). Several preprocessing techniques have also been applied to improve the performance of the model to Rescaling values back to the range [0,1].

Data Preprocessing Process:

1. The first step in this process is to normalize the pixel intensity to a range between 0 and 1 to normalize the data to achieve consistency in dynamic range for the input images (Koo, K. M., & Cha, 2017).
2. Data Augmentation using rotation, inversion, and zoom to simulate realistic conditions and reduce overfitting (LeCun, Y., Bengio, Y., & Hinton, 2015).
3. Batch processing was employed in this research to apply consistent preprocessing operations to multiple images simultaneously. The use of batching and random shuffling contributed to enhanced efficiency during the training phase (Barnaby, C., Chen, Q., Samanta, R., & Dillig, 2023).

Model Architecture:

CNN-based models are used efficiently in TSR systems to classify signs effectively (Alkhazraji, L., Abbas, A. R., Jamil, A. S., Kadhim, Z. S., Alkhazraji, W., Jebur, S. A., ... & Hussain, 2025). The system is based on a convolutional neural network (CNN) consisting of:

1. An input layer that resizes images
2. Three convolutional Layers followed by maximum pooling using filters (32, 64, 128)
   - Flatten layer to convert output to vector.
   - Dense layer (density) with 128 cells.
   - Softmax output layer with a number of cells equal to the number of sign varieties.

The Adam algorithm was used as an optimization tool (Optimizer), with a cross-entropy as loss function due to the multivariate nature of the problem.

Training and Evaluation Model:

The model was trained on 10 epochs, using test data to accurately assess performance. A built-in software interface was also used that displays the training progress instantaneously using Tkinter.

System Implementation and Graphical Interface:

An interactive graphical interface has been developed that allows the user to upload a photo of the traffic sign, and then display the result with the percentage of confidence, as well as make a sound in Arabic stating (indicating) the name of the sign.

An audio feed system using the pygame library, matches the name of the labeled sign with a pre-recorded audio file. As a summary, the proposed sign classification model's process consists of the following steps:

- Import necessary libraries such as TensorFlow, Tkinter, Pygame and others.
- Initialize the data paths and folders needed for the form and audio.
- Check for training and test data folders.
- Load images from data folders and divide the training data into two parts: Training and Verification.

- Extracting category names from training data for later use in classification and presentation.
- Build a model using MobileNetV2 as the base model with additional layers for classification.
- Preparing the model for training using Adam and loss sparse_categorical_crossentropy.
- Train the model on the data and display the training progress with a progress bar.
- Plot the accuracy and the loss of curves during training and verification phases.
- Predicting test data rankings.
- Calculate metrics (Accuracy, Precision, Recall, F1 Score) using the sklearn library.
- Display the results in the form of a heatmap and show a message with the metrics.
- Save the trained form in.keras format.
- Open a new image from the device and upload it inside the interface.
- Categorize the image using the trained model and demonstrate classification and confidence.
- Play the audio file associated with the classified category, if available.
- Display the image and result inside the user interface.
- Provide an option to save the performance report in CSV format when needed.
- Create a graphical interface using Tkinter containing buttons, progress bar, and display elements.
- Run the program and interact with the user using graphical user interface (GUI).

System and User Interface

This section provides a detailed description of the graphical user interface specifically designed for the Arabic Traffic Sign Recognition (TSR) system. The interface aims to offer an interactive and intuitive environment that facilitates user interaction with the system, with particular emphasis on enhancing accessibility through features such as voice pronunciation of traffic signs. Additionally, this section reviews the results obtained during the training and testing phases of the proposed model, highlighting the system's performance and classification efficiency.

GUI Overview:

The GUI was developed using the Tkinter library in Python, aiming to provide an easy-to-use interactive environment for managing the traffic light classification system. The interface incorporates several key components, each designed to fulfill specific functions:

- Image upload feature: This allows users to upload an image of a traffic light from their device for analysis and classification by the intelligent model.
- Clear display of results: Upon completion of the classification process, the interface presents the predicted traffic sign name along with the confidence score provided by the model, assisting users in assessing the accuracy of the classification.
- Training progress bar: During the model training phase, this bar continuously shows the progress over time, enabling real-time monitoring of the process and evaluation of its speed and quality.
- Voice pronunciation of results: To enhance interactivity, the system audibly pronounces the name of the classified sign in Arabic, facilitating user understanding without requiring them to read the text.

Collectively, these components enhance the traffic light classification system by making it more interactive, transparent, and user-friendly.

Interface interior design components:

The graphical user interface is designed in an organized and practical manner, and the design consists of the following parts:

- Main window: contains buttons for training the model, uploading images, displays results and updates during operation.
- Image display area: shows the uploaded image before the classification process starts.
- Training follow-up bar: reflects the progress of the model's training process via a real-time visual indicator.
- Audio Outputs: When the sign is successfully classified, the system produces an audio pronunciation of the sign in Arabic, enhancing accessibility for people with special needs.

Figures 3 and 4 show examples of the windows that appear to the user during implementation, providing ease of communication and interaction with the designed system.

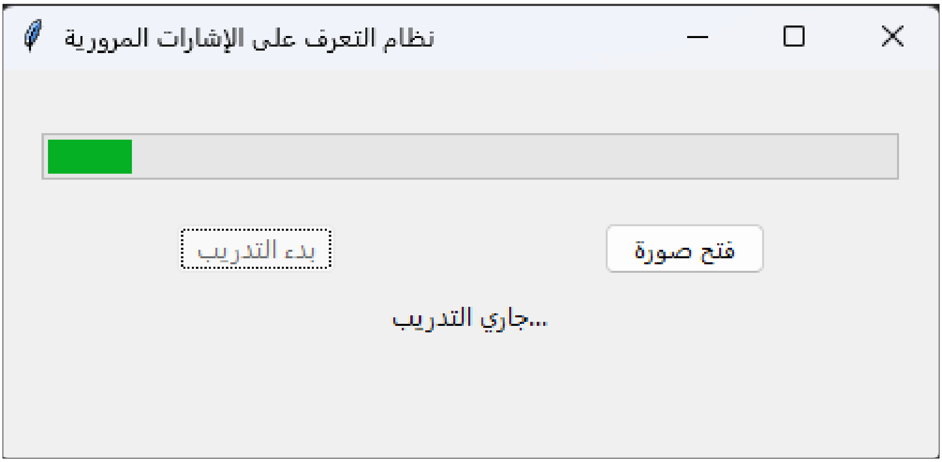
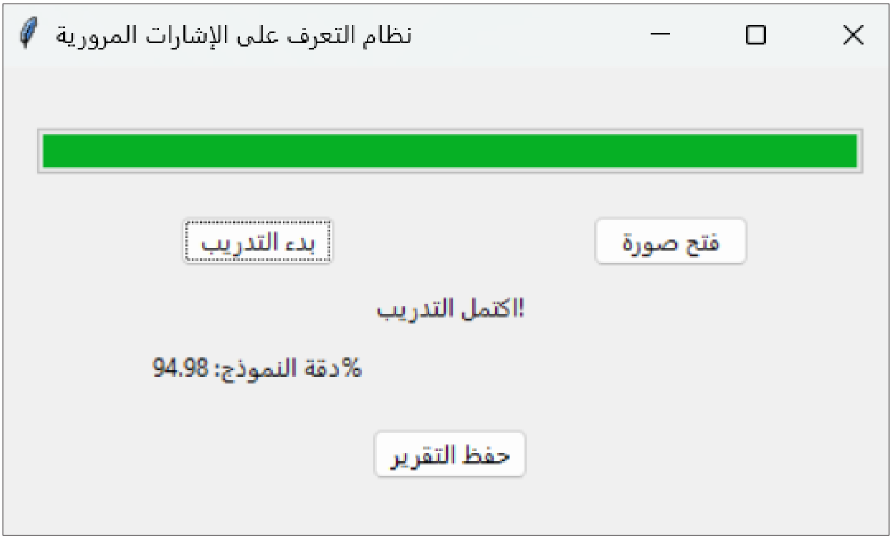


Figure 3: The initial interface, which displays the options to start training or upload a new image.



gure 4: Status of execution progress during model train.

Evaluation Results

The proposed model was tested on a dataset containing 3,000 images across 24 classes. Accuracy and other evaluation metrics, as listed in Table 2 (including accuracy, precision, recall, and F1 score), are essential for assessing how well the model's predictions align with the actual outcomes (Abdul-Jabbar, S. S., Farhan, A. K., Abdelhamid, A. A., & Ghoneim, 2022). These metrics are commonly used in classification tasks across AI applications and statistical analysis.

Table 2: The used metrics to measure the proposed model efficiency

Metric	Equation
Accuracy	(TP + TN) / (TP + TN + FP + FN)
Precision (PPV)	TP / (TP + FP)
Recall (Sensitivity, TPR)	TP / (TP + FN)
F1-Score (standard)	2 × (Precision × Recall) / (Precision + Recall)

The bypass neural network (CNN) model was trained using 3000 images spread over ten training cycles (Epochs). The test results showed the model's effectiveness and high accuracy, which confirms its efficiency in classifying Arab traffic signs. The performance of the model was also documented on a number of input images, and the accuracy ratios were as shown in Figure 5.

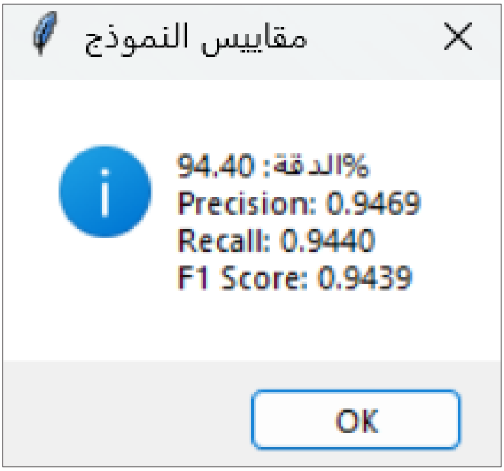


Figure 5: model test accuracy and other metrics.

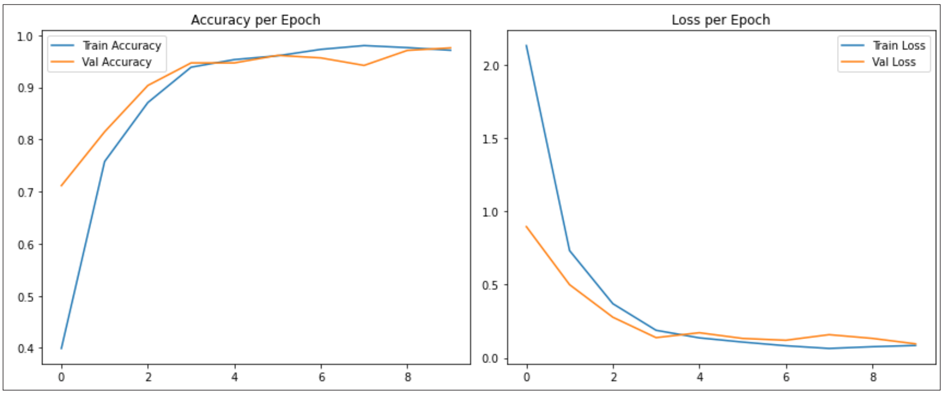


Figure 6: Training and validation performance curves

In Figure 6, the first diagram reflects the evolution of the model accuracy throughout the training and validation process. At the beginning of the era, training accuracy was relatively low, starting at about 40%, but it rose very quickly with each episode, indicating that the model was learning patterns and features in the data efficiently. By the fourth epoch, accuracy exceeded 95% and has roughly stabilized at 99% in recent epochs, indicating strong and stable learning. As for the accuracy of the verification, it started at a higher level than the accuracy of the training and this is sometimes normal when the verification set contains easier or clearer data and then it gradually increased until it became very close to the accuracy of the training, as it exceeded 97% in recent eras. This balance between the training and validation curve indicates that the model does not suffer from an overfitting problem, but rather retains its ability to generalize well to new data.

While, in the second graph, we note that loss began at high levels for both the training and verification group, but declined rapidly as the eras progressed. The training loss fell sharply and then stabilized at a very low level of near zero, reflecting the model's improvement in error reduction. Verification loss also followed a similar pattern, but remained slightly higher than training loss, with some slight fluctuations in recent eras. Despite these oscillations, the stability of the loss at a very low level indicates that the model not only learned patterns within the training set, but was also able to perform well on data it had not seen before.

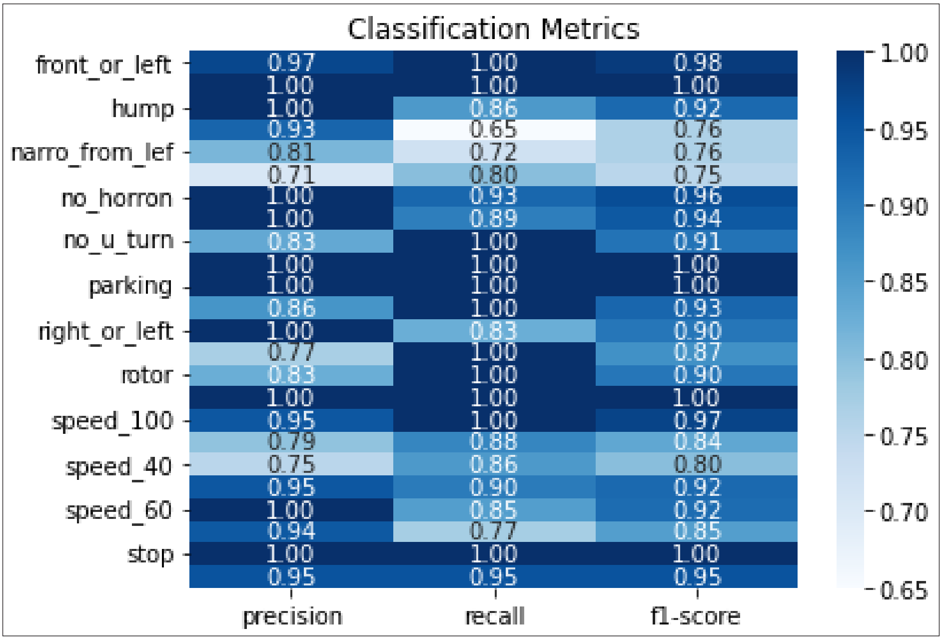


Figure 7: The Heat Map for the Classification Metrics During Model Training

On the other hand, Figure 7 presents a heat map that visualizes the performance metrics of the traffic sign classification model, including three key indicators: precision, recall, and F1-score. These metrics cover a variety of sign categories, such as directional signs (e.g., front_or_left, right_or_left), speed limits (e.g., speed_40, speed_60, speed_100), warning signs (e.g., no_u_turn, no_horn), and regulatory signs (e.g., stop, parking).

Overall, the heat map reveals that the model demonstrates robust and accurate performance, with most values ranging between 0.90 and 1.00. This indicates a high capability in distinguishing among the different sign categories. However, a few classes, such as narro_from_left, exhibited relatively lower performance, with recall and F1-scores around 0.72 and 0.76, respectively. This suggests that the model has some difficulty accurately recognizing these specific signs. In contrast, categories like no_u_turn and stop achieved perfect scores of 1.00 across all metrics, reflecting their visual clarity and ease of identification.

Based on the content of this visualization, an appropriate title for Figure 7 would be “Heat Map of Traffic Sign Classification Performance”, as it effectively captures the figure’s purpose and content.

Table 3: the valuation measures Precision, Recall, and the F1-score for 24 class of traffic signs

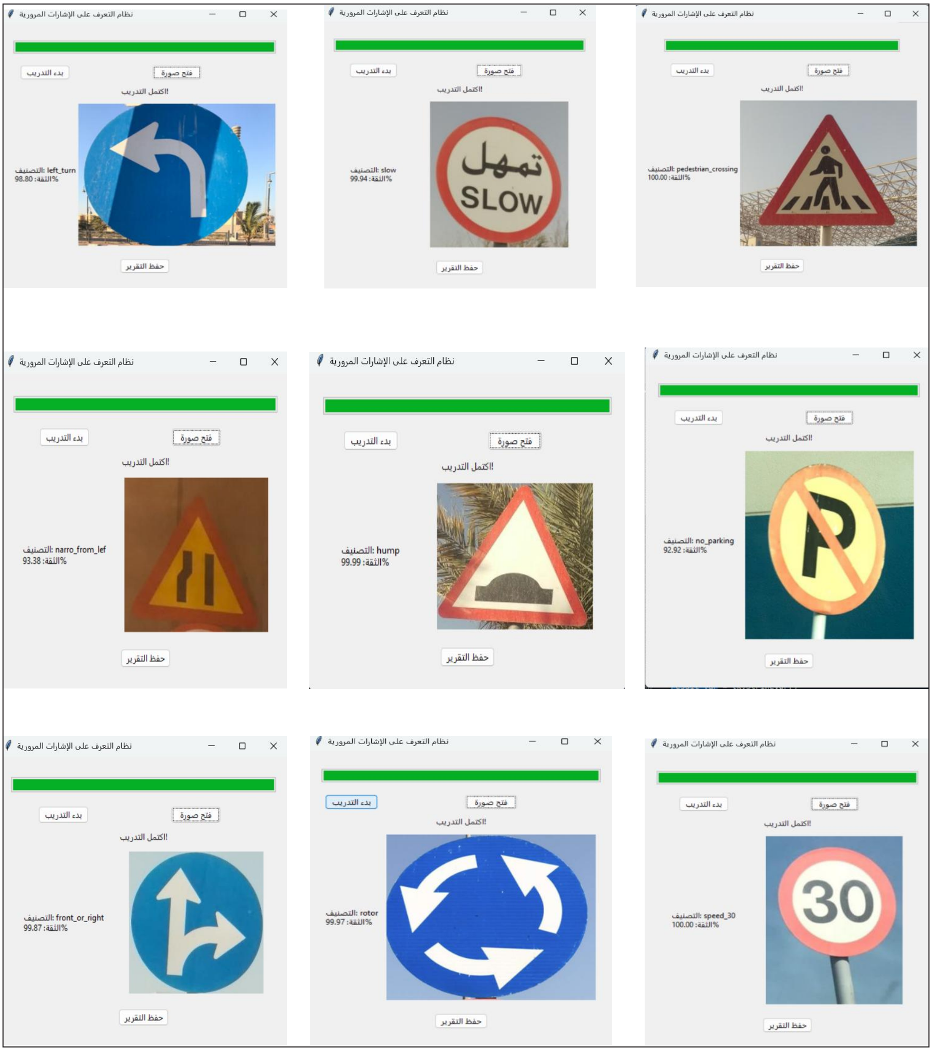
	precision	recall	f1-score	support
front_or_left	0.96875	1	0.984126984	31
front_or_right	1	1	1	30
hump	1	0.857142857	0.923076923	21
left_turn	0.928571429	0.65	0.764705882	20
narro_from_lef	0.8125	0.722222222	0.764705882	18
narrows_from_right	0.705882353	0.8	0.75	15
no_horron	1	0.925925926	0.961538462	27
no_parking	1	0.894736842	0.944444444	19
no_u_turn	0.833333333	1	0.909090909	20
overtaking_is_forbidden	1	1	1	22
parking	1	1	1	23
pedestrian_crossing	0.863636364	1	0.926829268	19
right_or_left	1	0.826086957	0.904761905	23
right_turn	0.769230769	1	0.869565217	20
rotor	0.826086957	1	0.904761905	19
slow	1	1	1	22
speed_100	0.947368421	1	0.972972973	18
speed_30	0.793103448	0.884615385	0.836363636	26
speed_40	0.75	0.857142857	0.8	21
speed_50	0.947368421	0.9	0.923076923	20
speed_60	1	0.85	0.918918919	20
speed_80	0.944444444	0.772727273	0.85	22
stop	1	1	1	21
u_turn	0.952380952	0.952380952	0.952380952	21
accuracy	0.916988417	0.916988417	0.916988417	0.916988417
macro avg	0.918444037	0.912207553	0.910888383	518
weighted avg	0.92494287	0.916988417	0.916774367	518

Similarly, the testing results shown in Table 3 include essential evaluation metrics—precision, recall, and F1-score—for each traffic sign category. The overall accuracy of the model is approximately 91.7%, which is a strong indicator of its capability to accurately classify traffic signs in most cases. Additionally, the weighted average F1-score is close to this value, signifying a good balance between precision and recall across all classes.

When analyzing individual categories, we observe several with outstanding performance—front_or_right, parking, stop, and slow—where all metrics reached a perfect score. This means the model did not misclassify any samples within these categories. Conversely, certain categories require improvement. For example, the left_turn class showed weaker performance, with a recall of only 65%, indicating that the model failed to detect this sign in several cases. Similarly, the narrows_from_right class displayed relatively low precision, pointing to confusion with visually similar signs.

Speed-related categories, such as speed_40 and speed_30, also showed inconsistent performance, likely due to visual similarity among the icons or varying image quality in the dataset.

From these findings, we can conclude that while the model performs very well overall, enhancements are needed for specific categories. To address this, techniques such as data augmentation, category rebalancing, or collecting additional training samples for underperforming classes could be beneficial.



ure 8: Various Examples of Final Classification Result When Entering an Image

The proposed model represents a qualitative step in the field of classification of Arab traffic signs, as the training and testing results obtained showed high accuracy that enables it to excel in this field, as shown in Figure 8. The success of the model was not limited only to its technical efficiency, but also to providing an improved user experience thanks to the integration of an interactive interface with voice feedback, which expands the scope of benefit, especially for drivers with special needs.

This system, based on deep learning techniques, is an effective tool that enhances the safety and efficiency of intelligent driving systems. High accuracy rates demonstrate the model's outstanding ability to generalize, ensuring reliable performance when dealing with new and diverse data. This qualifies the system to be a major part of the intelligent transportation system and traffic applications in the Arab world in the near future.

Overall, the results reflect the model's ability to achieve a masterful balance between accurately learning patterns and flexibility in adapting to unfamiliar data, confirming its feasibility and effectiveness in real-life applications related to Arabic traffic lights.

Conclusions and Future Works

In this research, an intelligent system based on deep learning techniques, specifically the convolutional neural network (CNN) model, was developed to recognize Arabic traffic signs during the Arbaeen visit. This system addresses challenges arising from the variety of sign designs, different fonts, and their linguistic characteristics. Additionally, an interactive graphical interface was designed to facilitate ease of use for visitors, incorporating Arabic audio feedback to enhance the user experience and make the system suitable for applications in autonomous vehicles and smart assistance systems.

Semi-Annual Scientific Journal c-karbala.com Vol.4, 4th year , Supplement (10)

Evaluation results demonstrated that the model achieved high accuracy (approximately 90% testing accuracy) in classifying traffic signs. However, some challenges remain, such as visual similarities between certain signs and the need for a more diverse dataset to enhance the model's generalizability. Accordingly, the research proposes several steps for future development, most notably expanding the dataset to include realistic images captured under varying environmental conditions to enhance model performance. It is also recommended to explore advanced network architectures such as ResNet, MobileNet, and Vision Transformers (ViTs), along with adopting transfer learning techniques to accelerate training and improve feature extraction quality.

From an implementation perspective, the model is expected to operate efficiently on embedded systems and mobile devices using tools such as TensorFlow Lite or NVIDIA Jetson, enabling real-time recognition in practical environments.

Moreover, the system's capabilities can be extended by integrating it with autonomous driving technologies and popular object detection frameworks such as YOLO or SSD to provide a more comprehensive understanding of the road scene. The integration of computer vision with multiple sensors like LiDAR and GPS through data fusion mechanisms is also a crucial step to enhance system performance in dynamic and complex environments.

Overall, this research provides a solid foundation for the development of intelligent transportation systems tailored specifically to Arabic environments, with broad potential to support traffic safety and improve intelligent interaction during large-scale events such as the Arbaeen Pilgrimage.

References

1. Abdul-Jabbar, S. S., Farhan, A. K., Abdelhamid, A. A., & Ghoneim, M. E. (2022). Razy: A String Matching Algorithm for Automatic Analysis of Pathological Reports. Axioms, 11(10), 547.
2. Alamri, S., & Kanwal, S. (2024). Enhancing Traffic Systems: Arabic Traffic Sign Recognition With Deep Learning. International Journal of Data Warehousing and Mining (IJDWM), 20(1), 1–18.
3. Alkhazraji, L., Abbas, A. R., Jamil, A. S., Kadhim, Z. S., Alkhazraji, W., Jebur, S. A., ... & Hussain, A. J. (2025). Employing the Concept of Stacking Ensemble Learning to Generate Deep Dream Images Using Multiple CNN Variants. Intelligent Systems with Applications, 200488.
4. Barnaby, C., Chen, Q., Samanta, R., & Dillig, I. (2023). Imageeye: batch image processing using program synthesis. Proceedings of the ACM on Programming Languages, 7(PLDI), 686–711.
5. Benfaress, I., Bouhoute, A., & Zinedine, A. (2025). Advancing Traffic Sign Recognition: Explainable Deep CNN for Enhanced Robustness in Adverse Environments. Computers, 14(3), 88.
6. Bhuvaneshwari, B., Abbijeet, R., Singh, S., & Priyadharshini, C. S. (2025). Traffic Sign Classification And Voice Alert System Using Convolutional Neural Network. Procedia Computer Science, 252, 995–1004.
7. Farzipour, A., Manzari, O. N., & Shokouhi, S. B. (n.d.). Traffic sign recognition using local vision transformer. In 2023 13th International Conference on Computer and Knowledge Engineering (ICCKE), IEEE, 191–196.
8. Fleyeh, H. (2004). Color detection and segmentation for road and traffic signs. In IEEE Conference on Cybernetics and Intelligent Systems, 2(IEEE), 809–814.
9. Khan, M. A., Park, H., & Chae, J. (2023). A lightweight convolutional

neural network (CNN) architecture for traffic sign recognition in urban road networks. Electronics, 12(8), 1802.

10. kolluri, o., Veldandi, S., & Pittala, S. (2024). Voice Based Assistance for Traffic Sign Recognition System Using Convolutional Neural Network. International Journal of Advance and Applied Research.
11. Koo, K. M., & Cha, E. Y. (2017). Image recognition performance enhancements using image normalization. Human-Centric Computing and Information Sciences, 7(1), 33.
12. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.
13. Manawadu, M., & Wijenayake, U. (2024). Voice-assisted real-time traffic sign recognition system using convolutional neural network. ArXiv Preprint ArXiv:2404.07807.
14. N G, G. K., Kishore, A., & Krishna, A. J. (2025). Real-time traffic sign recognition and autonomous vehicle control system using convolutional neural networks. Multimedia Tools and Applications, 1–36.
15. Ritter, W., Stein, F., & Janssen, R. (1995). Traffic sign recognition using colour information. Mathematical and Computer Modelling, 22(4–7), 149–161.
16. Wali, S. B., Hannan, M. A., Hussain, A., & Samad, S. A. (2015). An automatic traffic sign detection and recognition system based on colour segmentation, shape matching, and svm. Mathematical Problems in Engineering, 1(250461).
17. William, M. M., Zaki, P. S., Soliman, B. K., Alexsan, K. G., Mansour, M., El-Moursy, M., & Khalil, K. (2019). Traffic signs detection and recognition system using deep learning. In 2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS), IEEE, 160–166.