Bahrain’s Traffic Sign Recognition

Ahmed N Sadriwala

AIE 604: Deep Learning Applications

M.Sc. Student, University of Bahrain, Bahrain

# Abstract

As all auto manufacturers shift their focus to EVs (Electric Vehicles), the car is becoming more and more automated. Sensors are added to the vehicle to assist the driver during driving and to ensure the safety of the vehicle and its occupants. One of the sensors is the camera, which is typically installed in several different places throughout the vehicle. The camera is useful for lane management, distance management, and obstacle detection. This paper presents a You Only Look Once (YOLO) and onboard camera-based deep learning approach for the recognition of traffic signs. The suggested method enhances the performance of sign recognition by starting with a pre-trained YOLO model as a base network and fine-tuning it using a unique traffic sign dataset. The dataset includes a variety of shapes, colors, and symbols on annotated images of traffic signs from Bahrain. The YOLO model is then tested on a test set and found to be highly accurate and precise at locating and detecting traffic signs. The experimental results show the YOLO-based approach has potential for achieving cutting-edge performance in traffic sign recognition.

# 1. Introduction

Recognizing traffic signs is a crucial task in computer vision because it has a big impact on managing traffic and keeping roads safe. It's important to recognize traffic signs for Improving road safety, enhancing traffic management, supporting autonomous driving, and enabling smart transportation systems. Traffic sign recognition is crucial to these tasks.

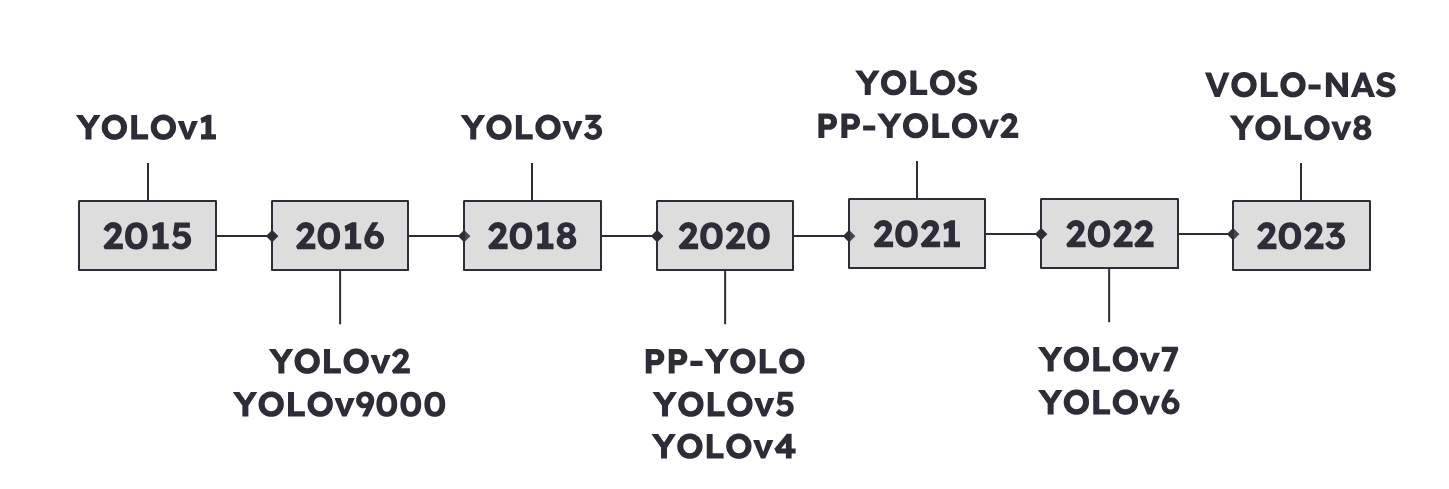
***Aim of the Project***

The goal of the project is identifying traffic signs of Bahrain using a Deep Learning model. At the end of the project a model will be produced to identify traffic signs from real time video streaming. We will be detecting the below signs:

* Several Speed Limit
* Stop
* Parking
* Road Hump
* No U-Turn
* Others

# 2. YOLO

YOLO proves to be cleaner and more efficient for doing object detection since it provides end-to-end training. Both the algorithms are fairly accurate but, in some cases, YOLO outperforms Faster R-CNN in terms of accuracy, speed and efficiency [6]. Below is the timeline for the release of all version of YOLO.



The bottom chart shows the comparison between all the YOLO models, and it is notice that YOLOv8 is superior then majority of the previous model.

A picture containing text, line, diagram, plot

Description automatically generated

The above chart is only valid for COCO dataset and using Nvidia T4 GPU. Depending on the dataset and the processing power the above graph may change. YOLO models are generally split into 3: small, medium, and large. Depending on the user requirement the model size needs to be selected.

## 2.1 YOLO-v8

YOLOv8 is the latest family of YOLO based Object Detection models from Ultralytics providing state-of-the-art performance.

Leveraging the previous YOLO versions, the YOLOv8 model is faster and more accurate while providing a unified framework for training models for performing:

* Object Detection.
* Instance Segmentation.
* Image Classification.

Below is the architecture of YOLOv8:

A screenshot of a computer

Description automatically generated with medium confidence

***Advantages***

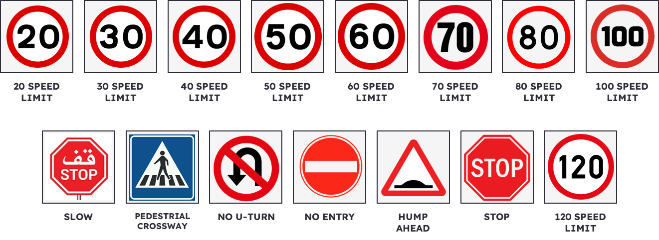
YOLOv8 has several advantages:

* Faster inference speed.
* Improved accuracy.
* Better generalization.
* It has a smaller model size compared to previous versions.
* Enhanced feature fusion.

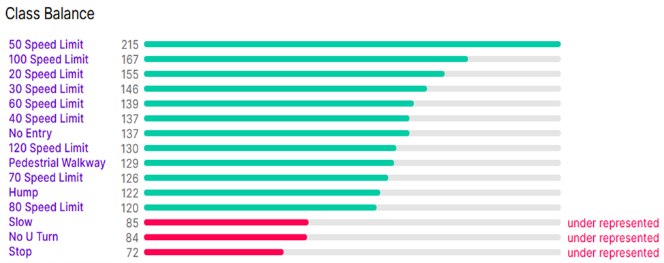
Overall, these improvements make YOLOv8 a more powerful and efficient object detection model than its predecessors, with better accuracy, speed, and generalization capabilities.

# 3. Dataset

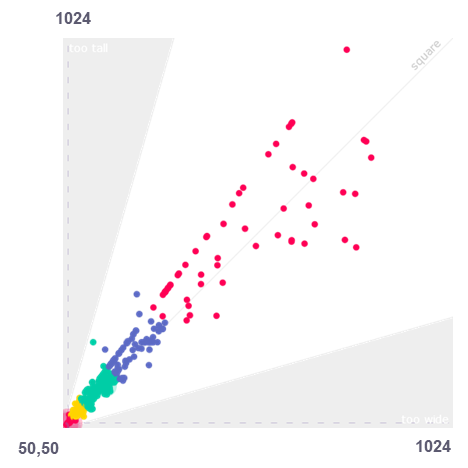
The dataset used for this experiment contains several images. Below are the classes of all the images:



There were 15 classes for all the images. The below chart shows the number of annotation for each classes.



In total, they were 1964 annotations from 1949 images. Each of this images had different sizes (resolution) which are shown below.



The data belongs from several different countries as they share the same traffic sign standards.

# 4. Tools Used

Several tool were used in this projects as mentioned below:

## 4.1 Roboflow

A computer vision platform called Roboflow offers the tools and infrastructure needed to create and use unique object detection and image segmentation models. The web-based interface for adding annotations to uploaded images, training models with cutting-edge algorithms, and deploying models to different platforms.

***Purpose in the experiment***

Roboflow is the most important tool used in this experiment as it was used for labeling all the images in the dataset. It is a web application where the dataset is uploaded, then classified bounding box is drawn on each image in the dataset which will produce the label for that image.

## 4.2 Visual Studio Code

Developers frequently use Microsoft's free and open-source Visual Studio Code (VS Code) code editor for coding and debugging. Python, JavaScript, C++, and many other programming languages are supported by it. Developers can customize their coding environment with extensions, themes, and other settings using VS Code's lightweight, adaptable interface.

***Purpose in the experiment***

Visual Studio Code is the application where the Python codes were written for real time use of YOLO model on video.

## 4.3 Google Colab

A free cloud-based service offered by Google called Colab (short for "Google Collaboratory") enables users to develop and run Python programs in a Jupyter notebook setting. It is a well-liked tool for tasks including deep learning, machine learning, and data analysis. Users can easily begin working on machine learning projects because it offers a virtual machine with pre-installed libraries and frameworks including TensorFlow, Keras, and PyTorch. Additionally, it gives users access to GPUs and TPUs, which are specialized hardware accelerators that may greatly speed up machine learning inference and training.

***Purpose in the experiment***

Google Colab was used to import and train YOLOv8 model on our custom dataset. It is also used to see the results of the training in terms of F1 score, loss, precision, and etc.

## 4.4 Python

Among the many industries that use Python are data science, web development, scientific computing, artificial intelligence, and more. Python is a high-level, interpreted programming language. It is renowned for being straightforward, readable, and simple to use. While still delivering sophisticated capabilities that make it a favorite among seasoned programmers, it has a clear and succinct syntax that makes it simple for newcomers to master.

***Purpose in the experiment***

Python was the main coding language used for importing, training, validating, and finally using the YOLO model on real time video streaming.

## 4.5 Ultralytics

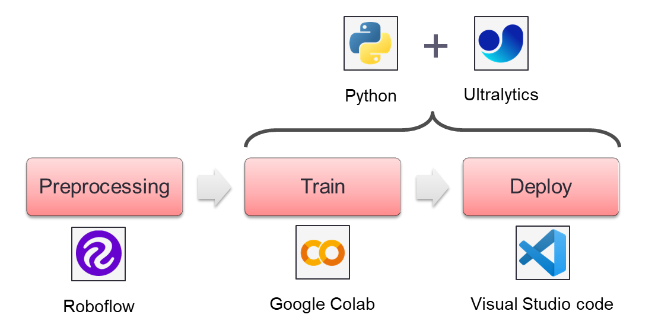
Deep learning and machine vision are the areas of expertise of the software company Ultralytics. The open source "YOLO" (You Only Look Once) object identification framework that the business created is well recognized for being a tool for object detection in pictures and videos.

***Purpose in the experiment***

Ultralytics is used in the experiment to import the YOLO v8 model and then using its API to train the model to the custom dataset.

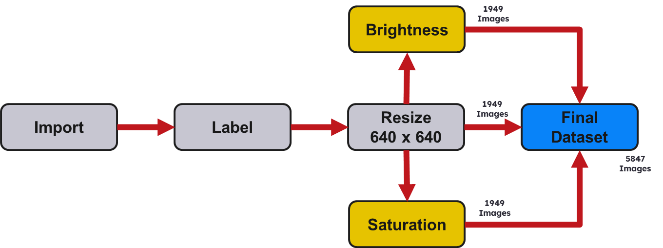
# 5. Experiment

In this experiment we will be importing a YOLOv8 model from Ultralytics and then later train it on our custom dataset which contains traffic signs from Bahrain. The project is split into main three parts which will be done in sequence and each part different tools will be used due to the advantages as shown below:



## 5.1 Pre-processing

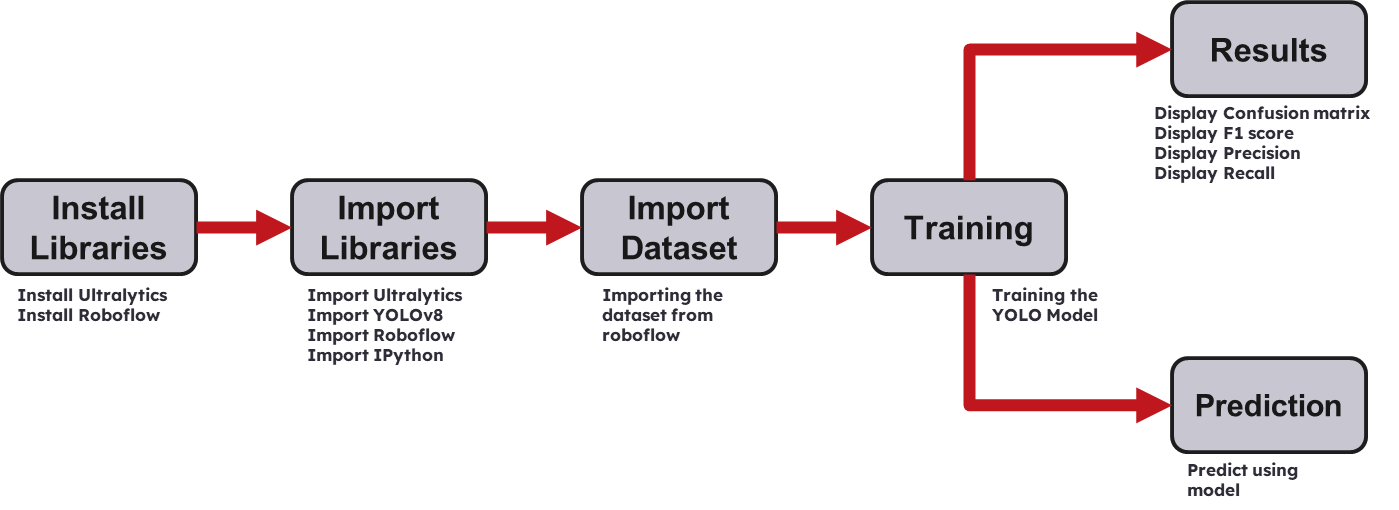
For the preprocessing stage an important tool is used i.e., Roboflow. Below is the flowchart of the process of preprocessing of data.



Firstly, the data imported into the Robo flow web application. Then manually each photo must go through labelling process, where the classification and the bonding box are defined for each image. Once done, the next step of preprocessing is to resize the image. As we know the default input of YOLOv8 is 640 x 640, hence all the images were resized to 640 x 640. Later augmentation process was done to increase the size of the dataset. The whole dataset was given random brightness change from +15 to -15. Also, the dataset went through random saturation changes from +25 to -25. With this process we have increased the size of the dataset by 3 times. Finally, the dataset is split into 70/28/2 for training / validating / testing respectively. Now the dataset is ready to be fed into the YOLOv8 model for training.

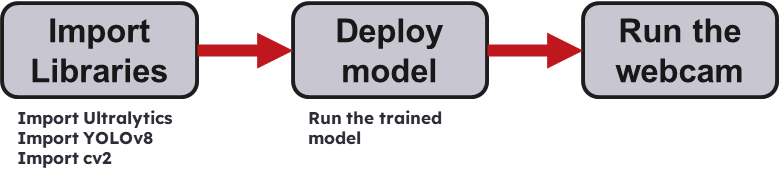
## 5.2 Train

Once the final dataset is ready it's time to train the model. Training the model is done in Google Colab to use Google provided GPU, to increase the speed of train. First we need to install all the necessary libraries which are Roboflow and Ultralytics. Once the installation of libraries is done, we will be importing those libraries. After importing those libraries, we need to import the dataset from Roboflow. The model is trained on the imported dataset on 50 epoch. The training took 4 hours and after the training results were produced in terms of confusion matrix, F1, recall, precision etc.



## 5.3 Deploy

To deploy the model on real time video streaming, first important libraries were installed such as Ultralytics (for the YOLO model) and CV2 (For video streaming). Then the model is Run using the trained weight from the previous section. Finally, the webcam was run, and the model computes each frame. Due to the processing power and model size, the program can only computer (i.e., one frame per 500 msec)



# 6. Results

Once the YOLO model is trained, several parameters need to be observed to see if the model is trained well and suitable to use.

## 6.1 Loss

There are main 3 losses to keep an eye while training a YOLO model

Box Loss: The box loss term measures the error in predicting the bounding box coordinates of the objects in the image.

Class Loss: The class loss term gauges how accurately the image's objects' class probabilities were predicted.

Distribution Focal Loss: The class imbalance problem in object detection, where there are many more negative examples (background) than positive examples (objects), is addressed using DFL loss.

***Training Data***

A picture containing text, diagram, line, plot

Description automatically generated

***Validating Data***

A picture containing text, diagram, line, plot

Description automatically generated

## 6.2 mAP

The average precision (AP) for each class of object in the image is first calculated, and the mean of the AP values for all classes is then calculated to produce mAP (Mean Average Precision).[6]

mAP50 (mean average precision at 50% IoU) is a commonly used evaluation metric that measures the accuracy of the model in detecting objects of interest in the image at a specific intersection over union (IoU) threshold of 50%.

mAP50-95 (mean average precision averaged across IoU thresholds from 50% to 95% with a step size of 5%) is a commonly used evaluation metric that measures the accuracy of the model in detecting objects of interest in the image across a range of intersection over union (IoU) thresholds.

A picture containing text, diagram, line, plot

Description automatically generated

## 6.3 Confusion Matrix

A confusion matrix is a table that compares the predicted and actual class labels of a set of test data to summarize how well a classification model performed. Because it demonstrates how easily the model can mistake one class for another, the matrix is given the name "confusion" [6].

***Training Data***

A screenshot of a computer

Description automatically generated with low confidence

***Validating Data***

A picture containing text, screenshot, diagram, line

Description automatically generated

In a typical confusion matrix, the predicted and actual class labels are represented by rows and columns, respectively, in a square table format. The matrix's diagonals show how many predictions were accurate, whereas off-diagonal elements show how many predictions were wrong [6].

## 6.4 Precision

For classification model evaluation, precision is a crucial metric, particularly when the cost of false positives is high. When a model makes positive predictions, precision is the percentage of those positive predictions that are accurate. A high precision score means that the model makes few false positive predictions, which means that many of its positive predictions are accurate. [6]

***Training Data***

A picture containing text, screenshot, plot, diagram

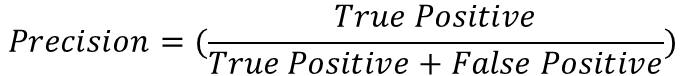
Description automatically generated

***Validating Data***

A picture containing text, screenshot, display, software

Description automatically generated

***Formula for Precision***



## 6.5 Recall

Recall is a measurement of how many actual positive cases, out of all possible positive cases, are correctly identified by a model. A high recall score shows that the model is thorough and correctly identifying many positive cases.[6]

***Training Data***

A picture containing text, screenshot, diagram, plot

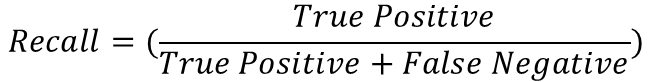
Description automatically generated

***Validating Data***

A picture containing text, screenshot, diagram, display

Description automatically generated

***Formula for Recall***



## 6.6 F1 Score

The effectiveness of the model in identifying objects of interest in the image is frequently assessed using the F1 score. A high F1 score means that the model is making few false positive and false negative predictions and is performing well in recall and precision.[6]

***Training Data***

A picture containing text, screenshot, diagram, plot

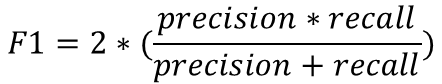
Description automatically generated

***Validating Data***

A picture containing text, screenshot, diagram, display

Description automatically generated

***Formula for F1 score***



## 6.7 Final Prediction

Below are test images from original dataset and model can recognize as seen:

A picture containing text, handwriting, circle

Description automatically generated

Below are actual Bahrain traffic sign images and model can recognize as seen:



# 6. Conclusion

Using the YOLOv8 object detection model, we presented a novel method for recognizing traffic signs in Bahrain in this paper. Our method makes use of YOLOv8's advantages, such as its high accuracy, real-time performance, and capacity for object detection at various scales and orientations. On a dataset of traffic sign images gathered from Bahraini streets, we trained and tested our model and saw encouraging results in terms of precision, recall, and F1 score. Our strategy could raise driving standards in Bahrain and elsewhere while lowering traffic violations and enhancing road safety.

The relatively small size of our dataset and the fact that we only considered a portion of the traffic signs frequently seen in Bahrain are some of the limitations of our study, though. To address these issues and investigate the generalizability and scalability of our methodology to different contexts and datasets, more research is required.

Our study lays the groundwork for future research on enhancing traffic sign recognition systems in Bahrain and elsewhere, and it adds to the growing body of knowledge on computer vision and object detection in the field of transportation.

# 7. Reference

[1] S. malik, “Traffic\_set,” Kaggle, https://www.kaggle.com/datasets/srthk5/traffic-set (accessed Jun. 9, 2023).

[2] HarelC, “Traffic signs dataset,” Kaggle, https://www.kaggle.com/datasets/harelc/traffic-signs-dataset (accessed Jun. 9, 2023).

[3] G. Latif, “Arts: Arabic traffic sign dataset,” Mendeley Data, https://data.mendeley.com/datasets/4tznkn45mx/1 (accessed Jun. 9, 2023).

[4] Xiaxxx, “Pieton,stop,50,30,speed\_bump,” Kaggle, https://www.kaggle.com/datasets/xiaxxx/geicar-pivoane (accessed Jun. 9, 2023).

[5] Model Structure of YOLOv8 Detection Models. Ultralytics, 2023.

[6] J. R. Terven and D. M. Córdova-Esparza, “A COMPREHENSIVE REVIEW OF YOLO: FROM YOLOV1 TO YOLOV8 AND BEYOND,” ACM COMPUTING SURVEYS.

[7] Effecient Frontier of Object Detection on COCO Measured on NVIDIA T4. Deci Research Team, 2023.

[8] M. M. Vikram, A YOLO Based Approach for Traffic Sign Detection, Apr. 2018. [1] C.

[9] Y. Wang and R. C. Yue, Traffic Sign Detection using You Only Look Once Framework.

[10] N. S. Artamonov and P. Y. Yakimov, “Towards real-time traffic sign recognition via YOLO on a mobile GPU,” Journal of Physics: Conference Series, vol. 1096, p. 012086, 2018. doi:10.1088/1742-6596/1096/1/012086

[11] D. V. Filatov, K. V. Ignatev, A. V. Deviatkin, and E. V. Serykh, “Application of machine learning methods for traffic signs recognition,” IOP Conference Series: Materials Science and Engineering, vol. 315, p. 012008, 2018. doi:10.1088/1757-899x/315/1/012008

[12] C. Gong, A. Li, Y. Song, N. Xu, and W. He, “Traffic sign recognition based on the YOLOv3 algorithm,” Sensors, vol. 22, no. 23, p. 9345, 2022. doi:10.3390/s22239345