



Conference Article

# Using AI-Powered Vehicle Identification System in Gas Stations (AI-VIS)

Uygar Usta<sup>1\*</sup>, Sumer Erkan Kaya<sup>2</sup>, Savas Baris<sup>3</sup>

<sup>1</sup> Asis Automation and Fueling System Inc., Istanbul, TURKIYE, Orcid ID: <https://orcid.org/0000-0003-4301-6830>, [u.usta@asis.com.tr](mailto:u.usta@asis.com.tr),

<sup>2</sup> Miggra Technology Inc., Istanbul, TURKIYE, Orcid ID: <https://orcid.org/0009-0001-3979-742X>,  
[sumer.kaya@miggra.com](mailto:sumer.kaya@miggra.com),

<sup>3</sup> Asis Automation and Fueling System Inc., Istanbul, TURKIYE, Orcid ID: <https://orcid.org/0009-0006-3135-8293>, [sbaris@asis.com.tr](mailto:sbaris@asis.com.tr),

\* Correspondence: [u.usta@asis.com.tr](mailto:u.usta@asis.com.tr); +90 531 380 73 74

(First received September 12, 2023 and in final form December 18, 2023)

**3rd International Conference on Design, Research and Development  
(RDCONF 2023)**

**December 13 - 15, 2023**

**Reference:** Usta, U., Kaya, S. E., Baris, S. Using AI-Powered Vehicle Identification System in Gas Stations (AI-VIS). *Orclever Proceedings of Research and Development*, 3(1), 255-263.

## Abstract

*In many countries around the world, retail fuel sales have to be recorded and monitored with specific vehicle information such as license plate by government institutions and station managers. Different hardware methods are utilized to achieve this goal such as UHF (Ultra high frequency) vehicle identification tags installed on the vehicles. To extract data from the tags, RFID-UHF antennas need to be installed on the nozzle for the recognition of vehicles today, which implies an increase in hardware costs per vehicle. Additionally, the electronic waste generated by the hardware used for vehicle recognition hurts the environment. In this study, the aim is to provide a comprehensive solution that enhances the modern automotive world's efficiency, security, and convenience. The core objective of this study is to design and implement a cutting-edge Vehicle Identification System (VIS) that leverages the power of Artificial Intelligence and Computer Vision. The proposed system has the ability to recognize various critical attributes of vehicles at gas stations, including the vehicle make, license plate, vehicle type, color, and fueling information.*



*The system utilizes advanced Image Processing and Deep Learning techniques to achieve precise identification and classification, improving security, and law enforcement.*

**Keywords:** Artificial Intelligence, AI, Vehicle Identification System

## 1. Introduction

Customer and vehicle recognition systems require specialized hardware and costly installation process for every vehicle in the network. Moreover, such hardware systems have been used extensively for the last 15-20 years, the capital costs have not seen a decrease as one would expect. Antenna installations require considerable downtime that results in loss of sales and are also costly to implement in gas stations due to specific AtEx (Atmosphere Explosives) certification requirements. In the end, they are limited to reading UHF tag ID. For example, hardware VIS systems cannot detect unregistered refills that are illegal in some countries (e.g., using containers) such as Turkey and UAE, identify hazardous objects (e.g., cigarettes), and are often late in reporting blacklisted vehicles. In today's world, it is imperative that the gas stations do not compromise security while serving millions every day. Therefore, relying solely on vehicle recognition systems creates security vulnerabilities. To mitigate rising costs and enhance security, systems need to be more complex and effective

With the advancement of Artificial Intelligence, it is now possible to design a system that not only recognizes vehicles at gas stations but also monitors fueling for security. This system is cost-effective and environmentally friendly, as it does not require additional per-vehicle hardware or costly installations. This study aims to design a state-of-the-art Vehicle Identification System (VIS) using the power of Artificial Intelligence and Computer Vision. This VIS system can identify critical vehicle attributes such as make, color, type, and license plate. In addition to vehicle attributes, it can also detect container usage, hazardous objects, and fueling information (hose, nozzle, fuel cap, container/jerrycan).

Deep Neural Networks, commonly used in computer vision applications, have the capability to solve complex tasks by learning patterns from input data [1]. They can be modified easily to solve challenges such as image recognition and object detection. A variant of Deep Neural Networks is Convolutional Neural Networks (CNN) which are used to solve image-driven pattern recognition [2]. With incremental progress in hardware and the availability of data, it has become easier to train Deep Neural Networks for image recognition tasks. VOC2007 object recognition challenge and ImageNet large-scale visual recognition challenge have both helped develop state-of-the-art CNN models



for object detection and recognition tasks [3], [4]. Image recognition refers to understanding the content of an image and categorizing it. Object detection identifies and locates specific objects or instances of objects within an image.

In this study object detection algorithm called YOLOv5 is used [5]. YOLOv5 is a CNN model that uses modern architectures for detecting objects in an image. YOLOv5 is not only very accurate but it is also very fast. The YOLOv5 family is comprised of five (5) models: YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x. YOLOv5 can also classify images, comprising 5 classification models: YOLOv5n-cls, YOLOv5s-cls, YOLOv5m-cls, YOLOv5l-cls, YOLOv5x-cls. As the network gets deeper from s to x, network accuracy increases while the speed of the model decreases. For this study, YOLOv5s and YOLOv5s-cls are used because they are fast and accurate. YOLOv5 is written in the PyTorch framework [6].

Recognizing license plates using only object detection is not enough, license plates may contain different fonts, sizes, and colors which are particularly hard to detect per character. Using object detection for recognition also involves time-consuming post-processing. For text recognition tasks, it is most suitable to use Optical Character Recognition (OCR) models. Optical Character Recognition in its simplest form, takes an input image with written text and outputs the written text as a string. To take advantage of this technology, a novel OCR model named PPOCRv3 is trained on a custom dataset of license plates [7]. PPOCRv3 is lightweight and very fast, easily fine-tuned for custom OCR tasks.

## 2. Materials and Methods

### 2.1. Hardware

For this study, the experiments were conducted using a custom-built desktop computer with the following hardware specifications:

- Central Processing Unit (CPU): An Intel Core i7-8700K processor, featuring 6 cores and 12 threads, with a base clock speed of 3.7 GHz and a turbo boost frequency of 4.7 GHz. The CPU was equipped with an Intel UHD Graphics 630 integrated GPU.
- Graphics Processing Unit (GPU): An NVIDIA GeForce RTX 2080 TI, a high-performance graphics card based on the Turing architecture, with 11 GB of GDDR6 VRAM. The GPU featured 4,352 CUDA cores, 68 RT cores, and 544 Tensor cores, providing substantial parallel computing power.



- Random Access Memory (RAM): A total of 32GB of DDR4 RAM, operating at 3200 MHz, configured in dual-channel mode. This configuration provided ample memory for data processing and simulations.

## 2.2. Dataset

For this study, four (4) distinct datasets were created. Part of the data was collected from the fuel station where the Proof of Concept (PoC) of this study was conducted, while the rest was gathered from the internet and publically open datasets [8] [9]. Annotation of the object detection dataset was done by drawing a bounding box around instances of classes [10].

These datasets are divided into four categories.

- Color recognition dataset (Classification)
- Make recognition dataset (Classification)
- Type (Commercial, Passenger, Motorcycle), fueling information (Hose, Nozzle, Fuel Cap, Container/Jerrycan), hazardous objects, license plate detection dataset (Detection)
- License plate recognition dataset (OCR)

*Table 1: Datasets Overview*

Dataset	Total Classes	Total Data
License Plate Recognition	36	17206
Make Recognition	36	109341
Color Recognition	14	203135
Type, Fueling-Information, License Plate (Combined)	10	2142



### 2.3. YOLOv5

Object detection and Classification is carried out using the YOLOv5 framework and separate models by their respective task.

YOLOv5 is an accurate and fast object detection algorithm developed by Ultralytics [11]. YOLOv5, short for "You Only Look Once version 5," is a single-stage object detection system and works seamlessly across multiple platforms. The YOLO architecture has undergone significant enhancements compared to previous iterations, making it a state-of-the-art choice for real-time object detection tasks [12].

**Input:** In YOLOv5, the input images are typically resized to a fixed size, commonly set at 640x640 pixels. It expects RGB color images, normalizes input data, employs data augmentation techniques such as mosaic augmentation, affine transformations, and color augmentations for training, and uses a multi-scale training strategy to adapt to different object sizes, making it suitable for real-time object detection in diverse applications.

**Backbone Network:** YOLOv5's backbone network is built upon CSPDarknet53, which is a modified version of the Darknet neural network architecture. The use of CSP (Cross Stage Partial Networks) enables more efficient feature reuse across different network depths, improving both speed and accuracy.

**Neck:** SPPF is a novel component integrated into YOLOv5. This component builds upon the concept of Spatial Pyramid Pooling (SPP) to capture multi-scale features from the backbone network. By employing SPPF, YOLOv5 improves its ability to detect objects of varying sizes within the same image.

**Head:** YOLOv5 uses YOLOv3 head for generating the final output. The detection head consists of multiple detection branches, each responsible for predicting bounding boxes, objectness scores, and class probabilities.

### 2.4. PPOCRv3

PPOCRv3 is used for recognizing vehicle plates. The text recognition algorithm of PPOCRv3 is used. For recognizing texts, PPOCRv3 introduces a lightweight text recognition network SVTR-LCNet [13].

SVTR-LCNet is a lightweight text recognition network that combines the advantages of transformer-based and CNN-based networks. It is based on the SVTR network, which is a state-of-the-art transformer-based network for text recognition. However, SVTR is relatively computationally expensive, making it unsuitable for real-time applications on mobile and embedded devices. SVTR-LCNet addresses this limitation by using a lightweight CNN backbone network, such as MobileNetV3, instead of the heavier ResNet



backbone network used in SVTR. This makes SVTR-LCNet significantly faster than SVTR while maintaining comparable accuracy. To train the model for text recognition, the CTC loss function is typically employed.

CTC loss is well-suited for sequence recognition tasks, as it accommodates variable-length sequences and is effective at aligning predicted sequences with ground truth data. SVTR-LCNet also includes several other optimizations to improve its efficiency, such as:

- A simplified decoder network
- A reduced number of encoder and decoder layers
- A knowledge distillation mechanism to transfer knowledge from a heavier and more accurate teacher model to a lighter and more efficient student model
- As a result of these optimizations, SVTR-LCNet is able to achieve state-of-the-art accuracy on a variety of text recognition benchmarks, while being significantly faster and more efficient than other state-of-the-art text recognition models.

## 2.5. TensorRT & Python

TensorRT was used to accelerate the inference of neural network models [14]. TensorRT reduced latency and increased throughput while reducing VRAM usage. Both C++ and Python were used for languages of choice. Python was used for training the neural networks and C++ was used for deployment.

## 2.6. Method

In this study, a pipeline consisting of YOLOv5 and PPOCRv3 was designed. For this pipeline, three YOLOv5 models and one PPOCRv3 text recognition model were trained. One model is responsible for detecting type, fueling information, hazardous objects, and license plate detection. One model for classifying the color and one model for the make of the vehicle were trained. For license plate recognition, the PPOCRv3 text recognition model was trained.

This study was conducted over a period of 3 months at 2 different stations, using a total of 8 cameras. Cameras communicated with the PC via wireless connection. During the testing period, 102,483 vehicles were detected, and 101,253 vehicles were successfully verified by the system.

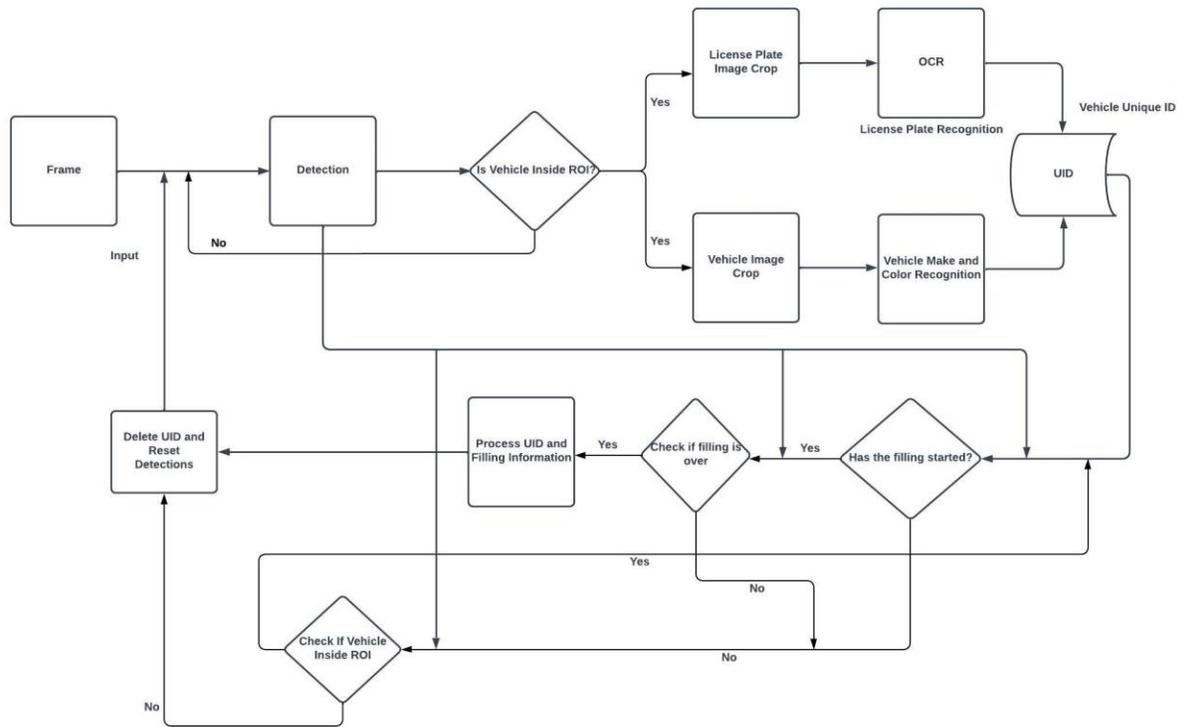


Figure 1: System Flowchart

### 3. Result

The results of the field tests conducted over a period of 3 months with 2 stations and 8 cameras are as follows.

- Devices operated 24/7 without any interruptions.
- No significant changes in performance were observed under varying lighting conditions during day and night.
- There were no disruptions in the wireless communication infrastructure used.
- Recognition of both square and rectangular license plates was successful.
- The time it took to recognize license plates as follows:
  - Minimum: 4.67 ms
  - Average: 5.52 ms
  - Maximum: 6.29 ms
- The success rate in detecting fuel transfer into a jerrycan or container could not be analyzed using data due to its limited sample size and lack of ground truth. Because there are legal restrictions to this mode of transfer in Turkiye, it is only limited to volume and cannot be recorded onto the automation software. However, over the course of 3 months,



the total number of detected fuel transfer operations into containers at both stations was a staggering 1,945. For this metric, a member of the team emulated this particular case to test the recognition. In each test, containers were successfully detected.

- Detecting hazardous objects could not be analyzed due to lack of sample size and ground truth. Synthetic data testing showed successful detection of cigarettes and knives.

*Table 2: Model and System Accuracies*

<b>Model</b>	<b>Accuracy</b>
License Plate Recognition	98.36 %
Make Recognition	98.23 %
Color Recognition	99.71 %
Type Detection	98.23 %
Fueling Information Detection	99.9 %
<b>System Accuracy</b>	<b>98.88 %</b>

In this study, NO DATA was stored in accordance with GDPR.

#### **4. Discussion and Conclusion**

The study encompassed a three-months-long field test conducted at two distinct gas stations utilizing eight explosion-proof cameras. The primary focus of the research was to evaluate the performance of the AI-Powered Vehicle Identification System in Gas Stations. During these tests, we examined the system's ability to accurately identify vehicle attributes, monitor fuel information, and detect hazardous objects. The result of this study shows that in the near future, using the power of artificial intelligence, monitoring vehicles in gas stations could be safer, cheaper and provide more insights.

For further improvements, recently developed algorithms like RTDETR, YOLOv8, and PPOCRv4 can be used to improve detection and recognition. Also, more classes can be integrated to detect and recognize various objects.

#### **5. Acknowledge**

In the scope of this paper, we would like to extend our gratitude to R&D Center of ASIS Automation and Fueling System Inc., and Miggra Technology Inc. for their contributions to the research.



## References

- [1] Deep learning with Python, F. Chollet
- [2] An Introduction to Convolutional Neural Networks, Keiron O'Shea and Ryan Nash
- [3] The Pascal Visual Object Classes Challenge, Mark Everingham Luc Van Gool
- [4] ImageNet Large Scale Visual Recognition Challenge, Olga Russakovsky Jia Deng Hao Su
- [5] [ultralytics/yolov5 – github](#)
- [6] Pytorch
- [7] PPOCRv3 More Attempts for the Improvement of Ultra Lightweight Ocr System
- [8] <https://platesmania.com/>
- [9] <https://www.kaggle.com/>
- [10] <https://www.makesense.ai/>
- [11] <https://www.ultralytics.com/>
- [12] [https://docs.ultralytics.com/yolov5/tutorials/architecture\\_description/](https://docs.ultralytics.com/yolov5/tutorials/architecture_description/)
- [13] PP-OCRv3: More Attempts for the Improvement of Ultra Lightweight OCR System
- [14] Nvidia TensorRT