Understanding the Accuracy and Reliability of Predicting Orientation of Cars in Autonomous Driving System

MSc Research Project
Data Analytics

Harshal Agashe
X22157051

School of Computing
National College of Ireland

Supervisor: Prof. Anu Sahani

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Harshal Agashe
Student ID:	X22157051
Programme:	Data Analytics
Year:	2023-34
Module:	MSc Research Project
Supervisor:	Prof. Anu Sahani
Submission Due Date:	14 Dec 2023
Project Title:	Understanding the Accuracy and Reliability of Predicting Orientation of Cars in Autonomous Driving System
Word Count:	3417
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Harshal Agashe
Date:	26th January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Understanding the Accuracy and Reliability of Predicting Orientation of Cars in Autonomous Driving System

Harshal Agashe

22157051

Abstract

Autonomous driving systems revolutionize transportation but face critical challenges in accurately interpreting their surroundings. A fundamental challenge is the precise determination of the orientation of surrounding vehicles, essential for collision avoidance and traffic flow optimization. This thesis addresses this challenge by investigating the effectiveness of the YOLOv8 model, a cutting-edge deep learning algorithm, in predicting vehicle orientations in varied autonomous driving scenarios. The research evaluates the model's performance in diverse environmental and traffic conditions, focusing on its accuracy and reliability. Using a comprehensive dataset from autonomous driving systems, the study conducts a thorough analysis of the YOLOv8 model, examining its capability to process and interpret complex image data. The findings aim to enhance understanding of the model's strengths and limitations in real-world applications, offering valuable insights for the development of more robust and efficient autonomous driving technologies. This work not only contributes to the field of autonomous vehicle research but also provides a framework for future advancements in vehicle orientation prediction, a critical component for the safety and efficacy of autonomous transportation systems.

Keywords: You Only Look Once (YOLO).

1 Introduction

1.1 Background

As the world moves towards the era of self-driving cars, precise prediction of automobile location and orientation is critical to assuring the safety and dependability of these revolutionary technologies (Nandy et al., 2020). In order to comprehend the complex dynamics of the road environment, autonomous driving technology depends significantly on sophisticated algorithms that process massive volumes of picture data (Ge et al., 2021). The accuracy with which these systems forecast vehicle location and orientation is critical to their ability to handle a variety of circumstances, from busy city streets to highways and varying weather conditions (Mo et al., 2016). The advancement and use of self-driving technology has been spectacular, with multiple businesses spending in research and development to bring these cars to the public (Nandy et al., 2020). However, difficulties exist, notably in making consistent accurate forecasts under dynamic and unpredictable real-world settings. According to Harris (2016), Google's autonomous vehicles in California encountered 272 failures between September 2014 and November 2015, and would have wrecked at least 13 times if human test drivers had not intervened. The effectiveness of image-based algorithms in determining vehicle location and orientation is impacted by factors like lighting conditions, the presence of other cars and pedestrians, and the responsiveness of the vehicle's control systems. This research tackles these issues by conducting a thorough analysis of the accuracy and dependability of forecasting automobile location and orientation in autonomous driving scenarios.

Figure 1: Fatal accidents damage trust in self-driving cars (Richter; 2018)

1.2 Significance

The research is significant since it evaluates the accuracy of forecasting automobile location and orientation, which is a vital part of autonomous driving. Its contribution lies in improving knowledge of the various aspects that influence prediction dependability. The research improves the underlying knowledge required for constructing safer and more efficient autonomous systems by focusing on the complexities of image-based algorithms under diverse conditions.

1.3 Research Aim

The aim of the research is to look at the accuracy and reliability of forecasting objects located in the image data with orientation in autonomous driving scenarios using image data.

1.4 Objective

- To evaluate the accuracy and reliability of detecting objects to predict the car orientation in autonomous driving scenarios using real-world scenarios.

1.5 Question

How does the accuracy and reliability of detecting objects to predict the car orientation in autonomous driving scenarios using real-world scenarios?

1.6 Outline of the Research

Chapters	Description
Introduction	Overview of the relevance of autonomous automobile prediction, as well as research problems for position and orientation accuracy
Related Work	Analysis of extant literature, studies, and technology concerning autonomous vehicle prediction
Research Methodology	Detailed strategy for data collection, analysis methodologies, and experimental design to meet the research objectives
Design Specification	Blueprint of the proposed models, algorithms, and systems for predicting car position and orientation
Implementation	Execution of the designed models and algorithms in a practical context using image data
Evaluation	Assessment of the accuracy and reliability of the predictions
Conclusion and Future Work	Summary of findings, implications, and future improvements in autonomous vehicle production systems

Table 1: Outline of the Research

2 Related Work

Khare and Jain (2023) provide a robust framework for predicting driverless automobile performance that integrates cognitive data analysis with reliability analysis, employing a decision tree, confusion matrix, and multi-regression analysis. Their primary focus is on evaluating and anticipating vehicle capabilities and dependability.

Gupta et al. (2021), on the other hand, provide a complete assessment of deep learning applications for object identification and scene perception in self-driving automobiles. The authors investigate the theoretical underpinnings of self-driving vehicles from a deep learning standpoint, with the goal of bridging the gap between deep learning and automobiles.

While Khare and Jain (2023) focus on practical prediction methods, Gupta et al. (2021) provide a broader overview of deep learning functions in self-driving technology. Thus, implementation of both methods can help in achieving higher results.

Rehrl and Gröchenig (2021) focus on the accuracy of autonomous driving systems' localisation, proposing approaches for validating vehicle position trajectories against various ground facts. Their findings show that this technique can achieve accuracy of less than 0.1m at confidence levels of up to 99.9%. Prédhumeau et al. (2022), in contrast, tackle the difficult challenge of forecasting pedestrian trajectories near autonomous cars. Their expert model combines the social force model with a decision model to generate real time, explainable trajectory predictions based on a variety of pedestrian behaviours. These findings can be used to develop methods for increasing the performance and safety of self-driving cars.

Djuric (2018) use deep learning and rasterized representations to anticipate uncertainty-aware short-term mobility for traffic actors in autonomous driving. They fulfil the requirement for safe and efficient operations by taking into account the inherent uncertainty in forecasting future movements. Jia et al. (2023) on the other hand, offer a quick and accurate object detector for autonomous driving based on enhanced YOLOv5. Through structural re-parameterization and neural architecture search, they improve the accuracy of the model and speed. While Djuric (2018) highlight uncertainty in anticipating traffic actor movements, Jia et al. (2023) focus on enhancing object identification accuracy and real-time performance. The research focuses on developing technology for safe and efficient autonomous driving. Both researchers contribute to the larger objective of improving technology that is required for the effective integration of autonomous cars into real-world contexts.

Wang et al. (2021) present a real time object identification system for autonomous cars based on YOLOv4 That balances detection speed and accuracy. They enhance accuracy by altering the backbone and neck of the YOLOv4, yielding an improvement in accuracy of 2.06% and 2.95% on the KITTI and BDD datasets, respectively. Cui et al. (2021), on the other hand, concentrate on uncertainty-aware estimates of vehicle orientation, which is critical for motion prediction in self-driving systems. Their strategy improves orientation inference and quantifies prediction uncertainty, yielding cutting-edge results on the dataset. Both Wang et al. (2021) and Cui et al. (2021) provide useful insights and approaches for addressing crucial issues of object identification and orientation estimation in autonomous cars.

Cao et al. (2022) emphasise the necessity of autonomous cars' decision-making systems, particularly in complicated urban traffic settings. The authors emphasised the need of improving decision making algorithms which improve the intelligent level of autonomous driving, and their proposed future integration of artificial intelligence approaches such as cognitive mapping and behaviour prediction. In contrast, Zaghari et al. (2020) use deep neural network approaches to improve self-driving car learning through Real world driving behaviour. Their LSV-DNN model, which uses the YOLO algorithm to identify obstacles, displays good learning of driver reactions. The findings of both the studies highlight the importance of machine learning algorithms and decision making for accuracy prediction in driverless cars.

The behaviour prediction function of autonomous vehicles usually estimates the future states of the nearby vehicles depending on the current and past observations of the surroundings. While Huang et al. (2023) concentrate on motion prediction, stressing improved designs in encoding, decoding and occupancy flow prediction, Mozaffari and Mouzakitis (2022) investigate behaviour prediction, emphasising its function in increasing vehicle ever nest of impending threats. The authors recognise the importance of deep learning in dealing with the complexities of autonomous driving scenarios. Huang et al. (2023) divide advancements into three categories such as scene input representation, context refinement and prediction rationality enhancement, whereas Mozaffari and Mouzakitis (2022) divide solutions into 3 categories such as input representation, output type and prediction technique. The studies agree on the importance of deep learning in tackling motion and behaviour prediction difficulties for autonomous cars, giving significant insights and possible research areas. Intelligent Automation (IA) in automobiles combines robotic process automation and artificial intelligence, allowing digital transformation in autonomous vehicles. Biswas et al. (2021) concentrate on focusing the future coordinates of various agents around autonomous vehicles such as automobiles, pedestrians and bicycles. They utilise the root mean square error (MSE) score to look at the adequacy of different deep learning models, with models involving the present status of the environmental elements as contribution to expect agent movement.Bathla et al. (2022), On the other hand, the Intelligent Automation (IA) in autonomous cars, which combines robotic process automation with artificial intelligence. They go into simulated intelligence, AI, and IoT approaches in AVs, analysing safety necessities, issues, and threats. The studies agree on the need of technology improvements for autonomous vehicles, with each carrying a unique technique to movement prediction and a detailed assessment of IA uses and issues. The efficiency and convenience of connected and autonomous vehicles have been increased as a result of the development of technologies like embedded devices (Liu et al.; 2023).

The extreme gradient boost (XGBoost) model is utilised by (Liu and Fan; 2021) to forecast vehicle direction in an connected and autonomous vehicle climate. The review thinks about the XGBoost model to the smart driver model (IDM) utilising Next Generation Simulation (NGSIM) datasets. The XGBoost Model outperforms the IDM in terms of prediction errors, with longitudinal position identified as the most important factor. In contrast, (Shibly and Kadobayashi; 2023) explained the security of connected and autonomous vehicles by presenting an adversarial attack protection method. The study underlines the susceptibility of connected and autonomous vehicles to tampering, which may result in dangerous driving circumstances. The authors provide an auto encoder and a compressive memory module to protect against adversarial inputs, and demonstrate its efficacy against various attacks on the Nvidia Dave 2 driving model.

Due to their global social, environmental, and economic benefits, autonomous vehicles have garnered attention recently. According to Hoermann and Dietmayer (2017), long-term probabilistic predictions of traffic participants are crucial for autonomous driving. Their probabilistic trajectory planning method treats inputs, outcomes, and parameters as random variables, minimizing computing work. Real-world traffic data shows that particle filter-based prediction can anticipate leading vehicles long-term. In contrast, Sharma and Indu (2022) addresses the challenging task of predicting pedestrian intentions for self-driving automobiles. Understanding pedestrian intentions is crucial for addressing safety concerns for vulnerable road users (Yannis and Jankowska-Karpa; 2020). A benchmark data set comparison is among the pedestrian intention prediction methods examined. It also discusses pedestrian intention prediction research challenges and solutions. The impact of improving autonomous vehicle capabilities and safety on busy public roadways is significant.

According to Biswas and Wang (2023), sensor technologies, mobile networks, and AI have enabled autonomous driving in various areas. Bharilya (2023) conducts considerable research on autonomous vehicle trajectory prediction, focusing on machine learning methods including deep learning and reinforcement learning. Bharilya (2023) analyzes over 200 publications, comparing methodologies and highlighting difficulties. The findings shed light on conventional and modern methodologies, datasets, and assessment criteria, advancing trajectory projection in autonomous vehicles. Malik et al. (2022) addresses autonomous vehicle decision-making fundamentals. They analyze decision-making strategies and approaches to use established methods. The research identifies decision-making complexity gaps and challenges that are essential for higher-level autonomous vehicle deployment. Thus, trajectory prediction and navigation are essential for self-driving cars. Comprehensive understanding improves autonomous vehicle safety and capability.

Autonomous vehicles have transformed the automobile industry, ensuring a better future when vehicles may drive without humans. Hubmann and Stiller (2017) discusses autonomous driving decision-making in dynamic and uncertain conditions. This Partly Observable Markov Decision Process (POMDP) considers other cars' concealed intentions. The approach entails improving an ego vehicle speed increase strategy for testing, trailing, and operational scenarios as given out on ensemble D2 using an intelligent, probabilistic movement model. This technology lets the autonomous vehicle adjust its approach based on forecast accuracy and future estimates. In contrast, P et al. (2022) discusses neural network topologies in the context of self-driving cars. The disruptive consequences of self-driving technology on transportation safety, efficiency, and human error are examined. The findings overcame profound learning techniques' exploratory results for explicit tasks including path planning, lane recognition, and traffic sign recognition. Increasing autonomous vehicle safety and capability makes the study important. One study explores choice complexity, while the other examines technology.

Predicting nearby vehicle behavior is crucial for autonomous vehicle monitoring and route planning. Kang and Chang (2020) employs a deep Social Generative Adversarial Network (GAN) model to predict self-driving car trajectory. By researching neighboring traffic agents' movement patterns and precisely predicting their future trajectories, autonomous autos can improve their decision-making. The suggested social-GAN method outperforms the classic social LSTM method, showing its applicability in complex traffic scenarios. Gao and Li (2022) emphasizes the need for autonomous automobiles to predict neighboring vehicle behavior over time. The hybrid model's movement recognition module captures contextual data using convolutional neural networks and long short-term memory. The behaviour prediction module uses LSTM and attention to forecast target vehicle behaviour from multi-time input. The approach has an average accuracy of 90% and a forecast time of less than 3 seconds. The research work together to improve autonomous vehicle trajectory prediction, each bringing valuable insights into different aspects of this difficult topic. Their combined impact on autonomous automobile safety and decision-making in varied traffic scenarios is enormous.

Existing research shows advancements in autonomous driving, with a focus on real time object identification, motion prediction and decision making in complicated traffic conditions. Despite significant improvements, existing systems frequently encounter issues in balancing speed and precision, effectively inferring vehicle orientation, and learning from real world driving behaviours. In order to solve these shortcomings and improve the overall capabilities of autonomous vehicles, innovative ways for forecasting the performance of driverless cars using cognitive data analysis and reliability analysis-based methodologies are required.

3 Research Methodology

3.1 Steps involved

Step 1: Using literature for theoretical knowledge

In order to distinguish the models helpful for object recognition through image data, a broad literature study was conducted. The applications and object detection methods of the YOLOv8 model were understood due to this.

Step 2: Data loading and null value check

The deployment of the YOLOv8 model includes various steps, which begins with the basic process of data loading. In this step, the essential datasets, such as images or frames for object detection, are first loaded onto the system. Post this, a thorough null value check is executed once the data is stacked in the system. This step guarantees integrity and completeness of the dataset.

Step 3: Applying YOLOv8 Model

This is the most crucial step which involves the application of the YOLOv8 model using the yolo command line utility to train the model.

Author	Aim	Method	Findings
Gupta et al. (2021)	To offer a comprehensive survey while bridging the gap between deep learning and self-driving cars	Deep learning models addressing real-time image perception challenges	100% accuracy on image classification and computer vision
Rehrl and Gröchenig (2021)	Address the absence of standardized process for verifying the precision of vehicle localization algorithms in automated driving systems	(1) A static driving path, (2) the lane center-line of a high-definition map, (3) localized vehicle body overlaps of lane boundaries in high-definition map; (4) longitudinal accuracy at stopping points, using two test data sets	Accuracy achieved above 95% confidence level for distance below 0.1m
Djuric et al. (2020)	Predict future traffic actor states for autonomous cars	Deep learning, raster pictures, convolutional models, and uncertainty consideration	Real-world trials and on-board testing for self-driving automobiles
Wang et al. (2021)	Improve the speed and accuracy of autonomous driving object identification	Improvements to the backbone and neck structure using a YOLOv4-based algorithm	Improves average accuracy on KITTI dataset by 2.06%, BDD dataset by 2.95%, speed of algorithm increased by 9.14%
Zaghari et al. (2020)	Create LSV-DNN for autonomous cars that mimics human driving behavior	Behavior cloning, real-world driving data, and YOLOv3 for obstacle detection carried out in Python and TensorFlow environment	High speed (more than 64.41%), low FPR (less than 6.89%), and low FNR (less than 3.95%)
Biswas and Wang (2023)	Examine the influence, implementation, and difficulties of IoT, EI, 5G, and Blockchain in AVs	Comprehensive survey of the literature on major enabling technologies for automated vehicles	Integration problems, future research paths, and opportunities
Hubmann et al. (2017)	POMDP to be developed for decision making in uncertain autonomous driving	The POMDP strategy for ego vehicle acceleration	The planning horizon reduces from 12s to 7.5s

Table 2: Summary of Research Papers

Step 4: Finding coordinates and confidence from the objects

After the YOLOv8 model has been deployed, the next step includes extracting coordinates and confidence ratings connected with the identified objects. These coordinates depict the position and size of the identified objects, while confidence percentages indicate the accuracy of each detection in the model.

Step 5: Detecting objects from the image

The final step includes the actual identification of the objects from the images. By applying the YOLOv8 model, several objects from the image dataset are identified.

3.2 Materials and equipment

The study includes a number of Python libraries such as Numpy, Pandas, Matplotlib, OpenCV, and PIL for Python Imaging Library. The libraries are used for different purposes; for example, Matplotlib is fundamental in data visualization with a variety of plots ranging from histograms to scatter plots(Hafeez and Sial; 2021). On the other hand, Numpy, which specializes in scientific computing, emerges as a critical component for image data prediction. The capability of this library is huge, multi-dimensional arrays and mathematical functions are useful for effectively managing and manipulating numerical data through Numpy (Nelli; 2015). Another library, Seaborn, which is usually built on Matplotlib, is highly useful for generating graphics based on the detected objects (Waskom; 2021). The capability of this model extends the capacity of Matplotlib through data visualization. Pandas add to the analysis of the study by giving data structures and operations appropriate for dealing with enormous datasets. The study likewise utilizes the OpenCV library which is appropriate for large computer vision libraries (Sivkov and Vasiliev; 2020). There is a broad collection of functions in OpenCV which helps in image handling and computer vision applications, subsequently making it reasonable for this research. Beyond these libraries, the study also utilizes Python Imaging Library or PIL for image handling activities, highly appropriate for simple manipulation of images ()(Rajab Asaad et al.; 2023). The utilization of glob modules contributes to efficient file path processing in this context.

3.3 Sample collection and preparation

A secondary data collection process has been used in this research. The required data was collected from Kaggle and this includes the images of self-driving cars Safurahajiheidari (2023). The random set of images of the self-driving cars from the dataset was used for this study which was uploaded in the model. The analysis technique for this research was quantitative which was performed through the implementation of the YOLOv8 Model. The null values in the dataset were checked and validated through the `isnull().sum()` method. Upon evaluation, it was observed that there were no missing values in the dataset, which demonstrated the integrity and completeness of the data for the model prediction.

3.4 Measurements and calculations

Extensive data preparation processes were performed to improve the overall quality of the dataset. The removal of null and duplicate values again increased the overall integrity and suitability of the dataset for model prediction. Besides, exploratory data analysis has been performed in order to identify the trends, patterns, and potential improvements in the model. Moreover, the images of the cars uploaded in the dataset provided a thorough understanding of the model. As all the images of car models were used, it offered a detailed evaluation and identification of objects.

3.5 Data Analysis techniques

The data analysis process involves the image recognition techniques that have been used for understanding the coordinates as well as the details of the self-driving cars. The analytical approach followed the CRISP-DM method, where certain steps were followed. The process initiated with understanding the data, preparing the data, further following the modeling, evaluation, and deployment of the YOLOv8 model. One of the most significant approaches of the data analysis technique is the implementation of the YOLOv8 model which helps in detecting images and objects. The YOLOv8 model has been trained by employing the validation method in order to evaluate the model against the grounded truth.

4 Design Specification

The design specification of this entire project mainly represents a comprehensive analysis of all the significant technique frameworks along with definite models employed in the implementation to predict car orientation in the case of autonomous driving by using specific image data, collected from Kaggle (Safurahajiheidari; 2023). The project aims to describe the final output of the implementation, involving the desired results produced in Python Programming language (Jupyter Notebook).

4.1 Identifying Techniques and Frameworks Utilized in this Project

The process of implementation eventually leverages diverse key frameworks and techniques that are described in this section with a detailed overview. The study initiates with loading and exploring all three CSV files such as Labels_train.csv, Labels_trainval.csv, and Labels_val.csv, including labeled bounding box information for car images. The Pandas Library is widely utilized to ensure effective data manipulation and analysis Raschka et al. (2020). This project also incorporates some crucial Python libraries and frameworks to adequately predict car orientation in autonomous driving systems. Key components involve NumPy, especially for numerical operations, Matplotlib is used for image preprocessing and visualization, OpenCV is incorporated for computer vision tasks, and the warnings library is used for enhancing the code reliability by eventually suppressing non-essential warnings Raschka et al. (2020). Ultralytics is used for importing the YOLOv8 model (yolov8m.pt), which aims to detect objects and segmentation with coordinates and class labels from images. Exploratory Data Analysis is then conducted with the Image data to visually explore the bounding boxes on the chosen image by using Matplotlib and PIL imaging libraries.

4.2 Description and Model Functionality of YOLOv8 Model

Autonomous driving systems handle diverse unexpected scenarios, and for that reason, applying a robust model in this project is one of the most significant approaches to control all the real-world conditions adequately. The YOLOv8 model is integrated into this project, which plays a central role in predicting car orientation. The model is a pivotal component in predicting all the significant orientations of cars from the image dataset. The YOLOv8 model is incredibly accurate for detecting segments as well as objects from an image Mudawi et al. (2023). The model is mainly known for its real-time object detection abilities by splitting the images into specific grids to predict bounding boxes, confidence scores, and coordinates or class labels Terven et al. (2023). Random images are loaded by using random to test the entire YOLOv8 model. In this project, 8 samples are randomly selected to ensure a representative and manageable dataset. This model is also applied to predict all the significant bounding box coordinates along with object labels and confidence sources, contributing to meaningful insights into the entire performance of the model. After finding the coordinates and confidence of each object from the randomly selected images, the objects are detected by overlaid bounding boxes. These boxes detect the representing objects with significant confidence values. The model functionality of the YOLOv8 model provides a crucial framework for predicting car orientation in the autonomous driving process by accurately detecting objects within images.

5 Implementation

5.1 Describing the Final Stage of Implementation

The primary aim of the final stage of implementation is to integrate as well as deploy the YOLOv8 model for detecting real-time objects for predicting the entire car orientation, especially for autonomous driving. This stage also contains the process of utilizing a dataset with specific annotation bounding boxes to train the YOLOv8 model. The desired outcomes mainly produce the successful execution of that trained model on a specific chosen set of test images. In addition, the model effectively objectifies cars within the selected images, offering all the specific bounding box coordinates, and confidence scores with specific class labels for all identified objects. Thus, this specific stage depicts the practical application of this entire study, encompassing its potential utility in various significant autonomous driving scenarios.

5.2 Description of Tools and Languages Used in this Project

The implementation procedure is mainly harnessed with various sets of tools and frameworks to accomplish the required output. Importing all the significant Python libraries is one of the most important steps to proceed with this entire project of predicting car orientation. Numpy and Matplotlib are widely used for numerical operations and image processing to provide a specific foundation, especially for data preprocessing and manipulation El Hachimi and Chehbouni (2022). These Python libraries also help in facilitating the exploration of the dataset, allowing for the bounding boxes and visualization of chosen images during the exploratory data analysis (EDA) step of this project. OpenCV is also one of the most powerful and significant computer vision libraries which is employed for

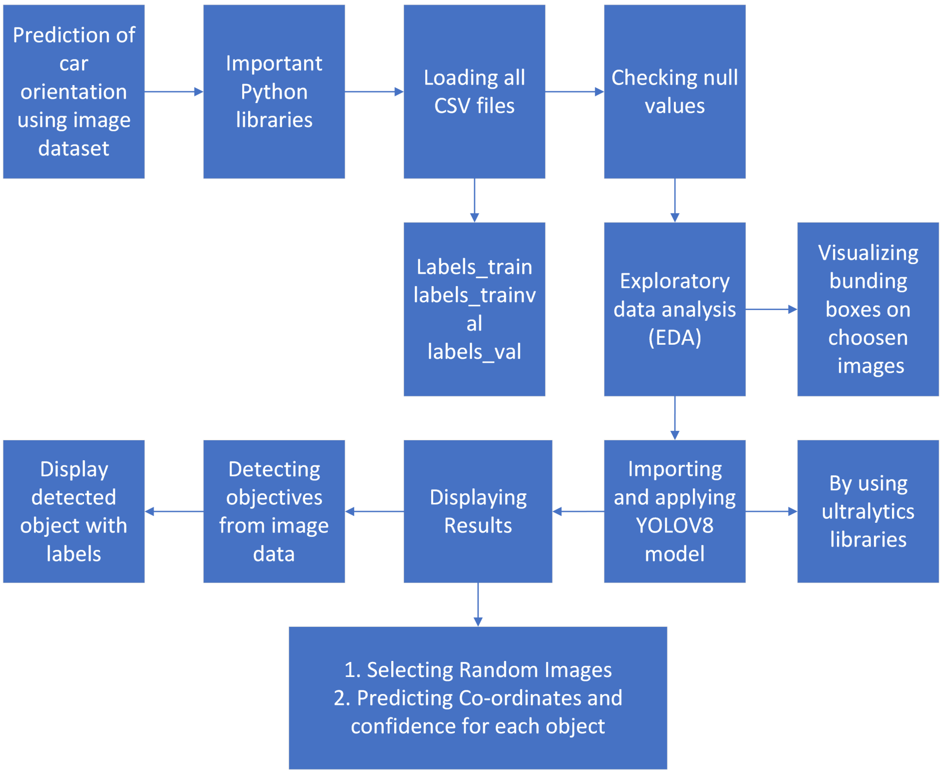

Figure 2: Diagram of the entire process

manipulation tasks, involving reading and displaying as well as resizing images Shi et al. (2022).

In addition, the incorporation of the YOLOv8 model within the context occurs by widely using the Ultralytics library, serving to identify each object present in the selected images. The implementation of all these libraries allows for a comprehensive exploration of the labeled data through EDA. Additionally, the Seaborn library helps to provide additional statistical graphical abilities for a clearer understanding of the overall distribution and characteristics of the image dataset.

Python served as one of the most efficient programming languages for this project of predicting car orientation, involving diverse significant tools and programming languages to ensure the accuracy and effectiveness of the integration of each functionality Raschka et al. (2020). The incorporation of the YOLOv8 model is achieved by using the Ultralytics library to streamline the entire integration process. This model also becomes the most important and effective part of detecting objects from the images to predict all the confidence scores and coordinates with class labels for each identified object in the selected images. Moreover, the entire study also employed Pandas in the software platform to handle as well as analyze the well-structured data. Three CSV files are also involved in labeled bounding box information which are loaded and followed by manipulation using these Pandas libraries. These facilitate a significant transition from car data to meaningful insights, providing insights into predicting the car orientation along with reliability and accuracy.

6 Evaluation

6.1 Importing Significant Libraries and Loading CSV Files

```
In [1]: import numpy as np
        import PIL
        from PIL import Image, ImageDraw
        from IPython.display import display
        import matplotlib.pyplot as plt
        from glob import glob
        import random
        import cv2
        import warnings
        warnings.simplefilter('ignore')

In [2]: #pip install opencv-python

In [3]: #!pip install ultralytics

In [4]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
```

Figure 3: Importing Important Libraries

Loading the CSV files

```
In [5]: df1 = pd.read_csv("labels_train.csv")
        df1.head()
```

Out[5]:

	frame	xmin	xmax	ymin	ymax	class_id
0	1478019952686311006.jpg	237	251	143	155	1
1	1478019952686311006.jpg	437	454	120	186	3
2	1478019953180167674.jpg	218	231	146	158	1
3	1478019953689774621.jpg	171	182	141	154	2
4	1478019953689774621.jpg	179	191	144	155	1

```
In [6]: df2 = pd.read_csv("labels_trainval.csv")
        df2.head()
```

Out[6]:

	frame	xmin	xmax	ymin	ymax	class_id
0	1478019952686311006.jpg	237	251	143	155	1
1	1478019952686311006.jpg	437	454	120	186	3
2	1478019953180167674.jpg	218	231	146	158	1
3	1478019953689774621.jpg	171	182	141	154	2
4	1478019953689774621.jpg	179	191	144	155	1

```
In [7]: df3 = pd.read_csv("labels_val.csv")
        df3.head()
```

Out[7]:

	frame	xmin	xmax	ymin	ymax	class_id
0	1478899046136829030.jpg	201	206	129	135	5
1	1478899046136829030.jpg	203	210	150	158	1
2	1478899046136829030.jpg	215	219	130	135	5
3	1478899046136829030.jpg	222	234	145	162	1
4	1478899046136829030.jpg	223	235	149	160	1

Figure 4: Loading the CSV Files

The project of predicting car orientation with accuracy and reliability in the autonomous driving industry involves image data, and for that reason, importing these Python libraries PIL, OpenCV, and Ultralytics are important. Numpy and Pandas are used for analyzing the accuracy and reliability during data preprocessing and model evaluation section, providing meaningful insights into the model performance Nguyen et al. (2023). Matplotlib and Seaborn help to enhance the interpretation of visualizations to ensure a robust autonomous system.

Figure 4 contains the entire process of loading CSV file names df1, df2, and df3, involving image labels for training and validation purposes. Each one of the rows of these CSV files represents a bounding box with frame, xmin, ymin, xmax, ymax, and class_id. These datasets enable the annotation to train the model for predicting car orientations by developing an accurate autonomous driving system.

6.2 Checking Null Values Present in the CSV Files

The three data frames df1, df2, and df3 have no null values, signifying a well-prepared dataset. The null values reduce the amount of knowledge learned by the ML models in the training section and affect the accuracy of the model negatively Palanivinayagam and Damaševičius (2023). Data cleanliness is essential to train robust models for accurately predicting car orientation in the field of autonomous driving systems. The absence of null values assures reliable learning, offering an effective system performance with safety.

Checking Null Values

```
In [8]: df1.isnull().sum()
Out[8]: frame       0
        xmin        0
        xmax        0
        ymin        0
        ymax        0
        class_id    0
        dtype: int64

In [9]: df2.isnull().sum()
Out[9]: frame       0
        xmin        0
        xmax        0
        ymin        0
        ymax        0
        class_id    0
        dtype: int64

In [10]: df3.isnull().sum()
Out[10]: frame       0
         xmin        0
         xmax        0
         ymin        0
         ymax        0
         class_id    0
         dtype: int64
```

Figure 5: Checking Null Values

Exploratory Data Analysis

```
In [8]: import os
import glob
import random
import cv2
import matplotlib.pyplot as plt

image_path = '/Users/harshal/Desktop/latest/images'

# Visualize bounding boxes
plt.figure(figsize=(15, 10))

# Set the desired range of images to display (2 to 3)
start_index = 2
end_index = 3

for i, row in df1.iterrows():
    # Break out of the loop if the desired range is reached
    if i + 1 > end_index:
        break
    if i + 1 < start_index:
        continue

    image_file = row['frame']
    image = Image.open(os.path.join(image_path, image_file))

    draw = ImageDraw.Draw(image)

    xmin, ymin, xmax, ymax = row['xmin'], row['ymin'], row['xmax'], row['ymax']
    class_id = row['class_id']
    draw.rectangle([xmin, ymin, xmax, ymax], outline='red', width=3)
    draw.text((xmin, ymin - 10), f'Class: {class_id}', fill='red')

    # Display the image
    plt.imshow(image)
    plt.axis('off')

plt.tight_layout()
plt.show()
```

Figure 6: Visualizing Bounding Boxes in a Randomly Chosen Image

6.3 Exploratory Data Analysis

The codes involved in figure 6 help to visualize all the bounding boxes on the selected image from the image dataset, encompassing the car positions for the autonomous driving system. The code displays random images within a specified range, containing Bounding Boxes with class IDs. The loop signifies through the rows of df1 dataset, including relevant information about images such as frame (image file name) and coordinates of bounding boxes (xmin, xmax, ymin, ymax) with class IDs.

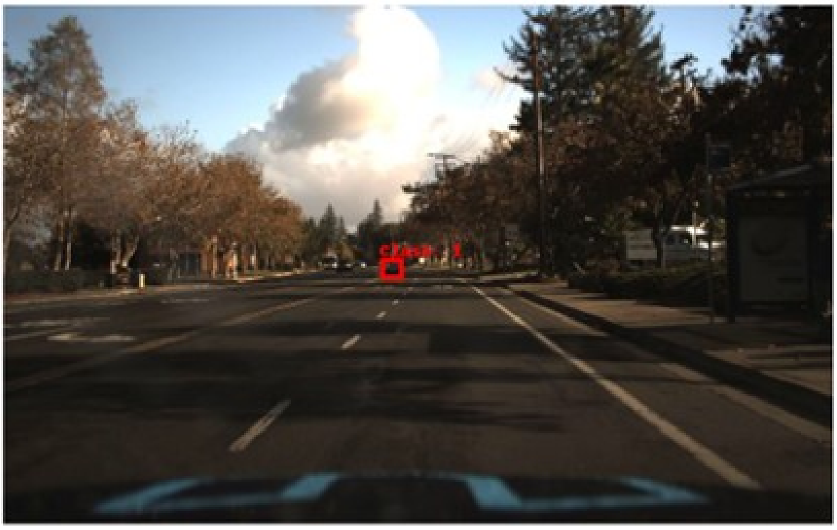

Figure 7: Output of Visualization of the Bounding Box

Figure 7 provides the resulting visualization of a clear representation of the Bounding Box, which is applied to the image by using the above-mentioned code. The outlined as well as labeled rectangles depict the understanding of the algorithm of car positions and aid in evaluating and further improving object detection reliability.

```
# Set the desired range of images to display (3 to 4)
start_index = 3
end_index = 4

for i, row in df1.iterrows():
    # Break out of the loop if the desired range is reached
    if i + 1 > end_index:
        break

    # Continue to the next iteration if the current index is less than the starting index
    if i + 1 < start_index:
        continue

    image_file = row['frame']
    image = Image.open(os.path.join(image_path, image_file))

    draw = ImageDraw.Draw(image)

    xmin, ymin, xmax, ymax = row['xmin'], row['ymin'], row['xmax'], row['ymax']
    class_id = row['class_id']

    # Draw bounding box
    draw.rectangle([xmin, ymin, xmax, ymax], outline='red', width=3)

    # Annotate with class_id
    draw.text((xmin, ymin - 10), f'Class: {class_id}', fill='red')

    # Display the image
    plt.figure(figsize=(5, 5))
    plt.imshow(image)
    plt.axis('off')

plt.show()
```

Figure 8: Displaying Class 1 and 2 within the Same Image

The above-mentioned codes of Python programming Language help to visualize images within the range of 3 to 4, contributing to a focused examination of significant frames in the image dataset. This code also assures the selection process of images within this range by adjusting the initial and end indices. The loop is created to load the image using the significant file path, specified in the df1 data frame. All the bounding boxes are eventually drawn around objects of interest. The rectangle is drawn in red color with a specific width of 3 pixels with class_ids, displayed above the bounding boxes.

The displayed images of Figure 9 provide the visualization of bounding boxes, serving as a targeted inspection of the df1 dataset, especially focusing on the frames between indices 3 and 4. This selective image assesses the accuracy of bounding boxes to understand the car positions in these two specific instances. The image subset evaluation process allows a detailed interpretation of the performance of the utilized algorithm in localized scenarios, providing a refinement as well as optimization of the autonomous driving system.

6.4 Evaluation of YOLOv8 Model

The code in Figure 10 illustrates the application and evaluation of the YOLOv8 model on a specific set of randomly selected images from the significant root path. This YOLOv8 model is mainly loaded by using the ultalytics Python library to enable effective object detection for the image dataset. The code begins by selecting a random subset of images, using num_samples (8) from the JPG images in the specified root_path. The code ensures that the selected number of samples falls within a desired range by thoroughly using min_num_samples and max_num_samples. The selected images are then displayed in a specific grid and arranged in two rows with a significant size of (20, 10).

Figure 9: Output of Displaying Class IDs

The output of these visualizations primarily illustrates the ability of the YOLOv8 model to detect objects such as cars, traffic lights, buses, and trucks within the randomly selected 8 images. These images represent a qualitative assessment of the performance of the YOLOv8 model, evaluating how well the model generalizes to real-world scenarios and refining the model's reliability in this context.

The output of exploring the coordinates and confidence of each object detected in the previously selected images indicates the presence of objects such as cars, trucks, persons, buses, and traffic lights with significant coordinates and confidence scores. The speed and accuracy of the model are illustrated by low interface timing and confidence scores, highlighting its potential for effective and reliable object identification in diverse scenarios. The exploration of coordinates and confidence is essential to assess the performance of the YOLOv8 model in identifying and localizing objects in the field of autonomous driving systems.

The code generates visualizations of images after detecting all the objects with class IDs and coordinates using the YOLOv8 model. The images are processed through the YOLOv8 model to display all the identified objects, which are highlighted by different colors. This representation is crucial for understanding the effectiveness of the model in

Applying YOLOv8 model

```
In [9]: import ultralytics
        from ultralytics import YOLO
        yolo_model = YOLO('yolov8m.pt')
```

Loading Images

```
In [10]: import os
import glob
import random
import cv2
import matplotlib.pyplot as plt

root_path = '/Users/harshal/Desktop/latest/images'
num_samples = 8
min_num_samples = 8
max_num_samples = 10


images_data = glob.glob(os.path.join(root_path, '*.jpg'))
random_images = random.sample(images_data, num_samples)
if num_samples < min_num_samples or num_samples > max_num_samples:
    print(f"Warning: num_samples ({num_samples}) is not within the desired range of {min_num_samples} to {max_num_samples}. Adjusting num_samples.")
    num_samples = min(max(num_samples, min_num_samples), max_num_samples)

if num_samples > len(images_data):
    print(f"Warning: num_samples ({num_samples}) is greater than the number of available images ({len(images_data)}). Adjusting num_samples.")
    num_samples = len(images_data)

plt.figure(figsize=(20, 10))
for i in range(num_samples):
    plt.subplot(2, max_num_samples // 2, i + 1)
    image = cv2.imread(random_images[i])
    plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB), aspect='auto', extent=(0, image.shape[1], 0, image.shape[0]))

    plt.axis('off')

plt.show()
```

Figure 10: Applying YOLOv8 and loading random images

Figure 11: Output of loading Random 8 images

```
image 1/1 /Users/harshal/Desktop/latest/images/1479502454739940895.jpg: 416x640 6 cars, 2 traffic lights, 570.7ms
Speed: 3.0ms preprocess, 570.7ms inference, 0.7ms postprocess per image at shape (1, 3, 416, 640)
Object 1 is: car
Coordinates are: [136.63650512695312, 88.30538177490234, 353.05438232421875, 267.0890808105469]
Confidence is: 0.91
-------
Object 2 is: car
Coordinates are: [346.0380554199219, 136.4490509033203, 452.56927490234375, 206.382080078125]
Confidence is: 0.9
-------
Object 3 is: car
Coordinates are: [56.912052154541016, 126.22277069091797, 145.8703155517578, 190.83175659179688]
Confidence is: 0.9
-------
Object 4 is: car
Coordinates are: [410.9279479980469, 121.86419677734375, 479.7298278808594, 251.4342498779297]
Confidence is: 0.89
-------
Object 5 is: car
Coordinates are: [0.23846054077148438, 111.99783325195312, 54.8818244934082, 238.83706665039062]
Confidence is: 0.82
-------
Object 6 is: car
Coordinates are: [33.74794387817383, 131.76873779296875, 72.27992248535156, 171.2501678466797]
Confidence is: 0.66
-------
Object 7 is: traffic light
Coordinates are: [384.891845703125, 86.97957611083984, 392.2591857910156, 105.04605102539062]
Confidence is: 0.62
-------
Object 8 is: traffic light
Coordinates are: [448.342529296875, 77.43537139892578, 454.32733154296875, 92.07728576660156]
Confidence is: 0.27
-------
```

Figure 12: Exploring coordinates and Confidence of each object

Detecting the objects

```
# plotting the images after object detection
print('\n\n------------------------------------ Images after object detection with YOLOv8 --------------------------------')

plt.figure(figsize=(20, 10))
for i, img in enumerate(images):
    plt.subplot(2, max_num_samples // 2, i + 1)
    plt.imshow(img)
    plt.axis('off')

plt.tight_layout()
plt.show()
```

```
------------------------------------ Images after object detection with YOLOv8 --------------------------------
```

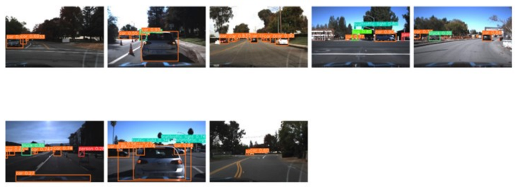

Figure 13: Detecting the objects within the randomly selected images

identifying diverse objects such as cars, traffic lights, trucks, and persons in real-world scenarios. The absence of axes and labels in the plotted images also focuses on the detected objects, transforming this into a concise and insightful representation of the model's performance in categorizing elements within the images.

6.5 Discussion

This section elucidates the significance of technological advancements in autonomous driving systems by evaluating the entire software analysis. Various existing research studies emphasize the importance of speed and accuracy in object identification in autonomous driving systems Wang et al. (2021); Zaghari et al. (2020). Based on the findings, the entire evaluation process of the YOLOv8 model demonstrates efficacy in real-world object detection scenarios through visualizing the bounding boxes with class IDs and coordinate scores of each object present in the image dataset. The exploration of bounding boxes along with class IDs also corresponds to the research study by Rehrl and Gröchenig (2021), emphasizing the validation of the positional trajectory of each vehicle, ensuring robust localization. The study incorporates diverse innovative methodologies and techniques, such as cognitive data analysis, to predict the car orientation of driverless vehicle systems. All the findings of this study affirm the evolving landscape of autonomous driving technologies, especially for advancements in safety and decision-making abilities.

7 Conclusion and Future Work

7.1 Conclusion

The project evaluates the YOLOv8 model for real-time object recognition and detection, illustrating its efficacy in enhancing the capabilities of the autonomous driving system. The study contributes to improving transportation infrastructure efficiency by detecting different objects with class IDs and coordinates. Integrating this entire data analysis with reliability analysis contributes to a broader landscape in predicting car orientation with improved dependability and accuracy.

7.2 Recommendation

Continual refinement of object detection algorithms, along with an enhanced focus on security perspectives, and collaborative efforts in standardizing the entire validation processes, are recommended for further improvements in autonomous driving technology. The findings of this study imply diverse potential avenues to address unavoidable challenges in autonomous systems and improve decision-making systems.

7.3 Future Scope

The future of autonomous driving research could explore significant advancements in edge computing for real-time processing systems and leverage emerging technologies such as 5G networks and blockchain technologies. Additionally, the incorporation of explainable decision-making algorithms or neural networks would help address promising areas in autonomous systems for further investigation. This evolution makes this dynamic field of autonomous vehicles more efficient and adaptable to real-world complexities.

References

Bathla, G., Bhadane, K. V., Singh, R. P., Kumar, R., Aluvalu, R., Krishnamurthi, R., Kumar, A., Thakur, R. and Basheer, S. (2022). Autonomous vehicles and intelligent

automation: Applications, challenges, and opportunities.
URL: *https://doi.org/10.1155/2022/7632892*

Bharilya, V. (2023). Machine learning for autonomous vehicle's trajectory prediction: A comprehensive survey, challenges, and future research directions.
URL: *https://doi.org/10.48550/arXiv.2307.07527*

Biswas, A. and Wang, H.-C. (2023). Autonomous vehicles enabled by the integration of iot, edge intelligence, 5g, and blockchain.
URL: *https://doi.org/10.3390/s23041963*

Biswas, S., Bianchini, M., Shaw, R. N. and Ghosh, A. (2021). Prediction of traffic movement for autonomous vehicles.
URL: *https://doi.org/10.1007/978-981-16-0598-7*$_1$2

Cao, Y., Chen, Y.-Y. and Liu, L. (2022). Research prospect of autonomous driving decision technology under complex traffic scenarios.
URL: *https://doi.org/10.1051/matecconf/202235503031*

Djuric, N. (2018). Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving.
URL: *https://doi.org/10.48550/arXiv.1808.05819*

El Hachimi, C., B. S. K.-S. and Chehbouni, A. (2022). Data science toolkit: An all-in-one python library to help researchers and practitioners in implementing data science-related algorithms with less effort.
URL: *https://doi.org/10.1016/j.simpa.2022.100240*

Gao, X., C. L. T. K.-C. Y. W.-H. and Li, Y. (2022). Deep learning-based hybrid model for the behaviour prediction of surrounding vehicles over long-time periods.
URL: *https://doi.org/10.1049/itr2.12214*

Gupta, A., Anpalagan, A., Guan, L. and Khwaja, A. S. (2021). Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues, *Array* **10**: 100057.
URL: *https://www.sciencedirect.com/science/article/pii/S2590005621000059*

Hafeez, A. and Sial, A. (2021). Comparative analysis of data visualization libraries matplotlib and seaborn in python.
URL: *https://doi.org/10.30534/ijatcse/2021/391012021*

Hoermann, S., S. D. and Dietmayer, K. (2017). Probabilistic long-term prediction for autonomous vehicles$_2$017.
URL:*https://doi.org/10.1109/ivs.2017.7995726*

Huang, R., Zhuo, G., Liu, X., Lu, S. and Tian, W. (2023). A review of deep learning-based vehicle motion prediction for autonomous driving.
URL: *https://doi.org/10.3390/su152014716*

Hubmann, C., B. M. A. D.-L. D. and Stiller, C. (2017). Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles$_2$017.
URL:*https://doi.org/10.1109/IVS.2017.7995949*

Jia, X., Tong, Y., Qiao, H., Li, M., Tong, J. and Liang, B. (2023). Fast and accurate object detector for autonomous driving based on improved yolov5.
URL: *https://doi.org/10.1038/s41598-023-36868-w*

Kang, L., H. C. W. I.-S. L. T.-L. C. S. and Chang, C. (2020). Vehicle trajectory prediction based on social generative adversarial network for self-driving car applications$_2$020.
URL:*https://doi.org/10.1109/is3c50286.2020.00133*

Khare, V. and Jain, A. (2023). Predict the performance of driverless car through the cognitive data analysis and reliability analysis based approach.
URL: *https://doi.org/10.1016/j.prime.2023.100344.*

Liu, H., Gong, L., Chen, X., Zheng, X. and Lv, C. (2023). Evaluating efficiency of connected and autonomous vehicles with different communication topologies.
URL: *https://doi.org/10.3390/electronics12173584*

Liu, P. and Fan, W. (2021). Extreme gradient boosting (xgboost) model for vehicle trajectory prediction in connected and autonomous vehicle environment.
URL: *https://doi.org/10.7307/ptt.v33i5.3779*

Malik, S., Khan, M. A., Elsayed, H. A., Khan, M. J. and Ullah, O. (2022). How do autonomous vehicles decide?
URL: *https://doi.org/10.3390/s23010317*

Mozaffari, S., A.-J. O. D.-M. J. P. and Mouzakitis, A. (2022). Deep learning-based vehicle behavior prediction for autonomous driving applications: A review$_2$022.
URL:*https://doi.org/10.1109/tits.2020.3012034*

Mudawi, N. A., Qureshi, A. M., Abdelhaq, M., Alshahrani, A., Alazeb, A., Alonazi, M. and Algarni, A. D. (2023). Vehicle detection and classification via yolov8 and deep belief network over aerial image sequences.
URL: *https://doi.org/10.3390/su151914597*

Nelli, F. (2015). The numpy library.
URL: *https://doi.org/10.1007/978-1-4842-0958-5*$_3$

Nguyen, L. P., Tung, D. D., Nguyen, D. T., Le, H. N., Tran, T. Q., Van Binh, T. and Pham, D. T. N. (2023). The utilization of machine learning algorithms for assisting physicians in the diagnosis of diabetes.
URL: *https://doi.org/10.3390/diagnostics13122087*

P, S. R., Gudla, R., Telidevulapalli, V. S., Kota, J. S. and Mandha, G. (2022). Review on self-driving cars using neural network architectures.
URL: *https://doi.org/10.30574/wjarr.2022.16.2.1240*

Palanivinayagam, A. and Damaševičius, R. (2023). Effective handling of missing values in datasets for classification using machine learning methods.
URL: *https://doi.org/10.3390/info14020092*

Prédhumeau, M., Mancheva, L., Dugdale, J. and Spalanzani, A. (2022). Agent-based modeling for predicting pedestrian trajectories around an autonomous vehicle.
URL: *https://doi.org/10.1613/jair.1.13425*

Rajab Asaad, R., Ismael Ali, R., Arif Ali, Z. and Ahmad Shaaban, A. (2023). Image processing with python libraries, *Academic Journal of Nawroz University* **12**(2): 410–416.
URL: *http://dx.doi.org/10.25007/ajnu.v12n2a1754*

Raschka, S., Patterson, J. and Nolet, C. (2020). Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence.
URL: *https://doi.org/10.3390/info11040193*

Rehrl, K. and Gröchenig, S. (2021). Evaluating localization accuracy of automated driving systems.
URL: *https://doi.org/10.3390/s21175855*

Richter, F. (2018). Infographic: Fatal accidents damage trust in autonomous driving.
URL: *https://www.statista.com/chart/13450/perceived-safety-of-self-driving-cars/*

Safurahajiheidari (2023). Yolov8 object detection on self-driving-car data.
URL: *https://www.kaggle.com/code/safurahajiheidari/yolov8-object-detection-on-self-driving-car-data*

Sharma, N., D. C. and Indu, S. (2022). Pedestrian intention prediction for autonomous vehicles: A comprehensive survey.
URL: *https://doi.org/10.1016/j.neucom.2022.07.085*

Shi, C., Funabiki, N., Huo, Y., Mentari, M., Suga, K. and Toshida, T. (2022). A proposal of printed table digitization algorithm with image processing.
URL: *https://doi.org/10.3390/a15120471*

Shibly, K.H., H. M. I. H.-T. Y. and Kadobayashi, Y. (2023). Towards autonomous driving model resistant to adversarial attack.
URL: *https://doi.org/10.1080/08839514.2023.2193461*

Sivkov, S., N. L. R. G.-R. A. V.-D. V. M. and Vasiliev, S. (2020). The algorithm development for operation of a computer vision system via the opencv library.
URL: *https://doi.org/10.1016/j.procs.2020.02.193*

Terven, J. R., Córdova-Esparza, D.-M. and Romero-González, J.-A. (2023). A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas.
URL: *https://doi.org/10.3390/make5040083*

Wang, R., Wang, Z., Xu, Z., Wang, C., Li, Q., Zhang, Y. and Li, H. (2021). A real-time object detector for autonomous vehicles based on yolov4.
URL: *https://doi.org/10.1155/2021/9218137*

Waskom, M. (2021). seaborn: statistical data visualization.
URL: *https://doi.org/10.21105/joss.03021*

Yannis, G., N.-D. L. A.-S. Y. B.-I. and Jankowska-Karpa, D. (2020). Vulnerable road users: Cross-cultural perspectives on performance and attitudes.
URL: *https://doi.org/10.1016/j.iatssr.2020.08.006*

Zaghari, N., Fathy, M., Jameii, S. M., Sabokrou, M. and Shahverdy, M. (2020). Improving the learning of self-driving vehicles based on real driving behavior using deep neural network techniques.
URL: *https://doi.org/10.20944/preprints202001.0283.v2*