CAR MODEL RECOGNITION using YOLOv8 ALGORITHM

A PROJECT REPORT

Submitted in partial fulfillment of requirement
for the degree of

Bachelor of Engineering
in
Electronics & Telecommunication

Submitted by

DIGVIJAY PATIL (B190953048)
YASH RANE (B190953050)
RISHAB KAR CHAUDHURI (B190953053)

Under the Guidance of

PROF. PRASHANT AHIRE

INNOVATION & LEADERSHIP
www.isquareit.edu.in

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION
HOPE FOUNDATION'S
INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY,
HINJAWADI, PUNE(MH)-411057
SAVITRIBAI PHULE PUNE UNIVERSITY

A.Y. 2023-24

CERTIFICATE

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION

HOPE FOUNDATION'S

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY,

HINJAWADI, PUNE-411057

This is to certify that

Digvijay Patil (B190953048)

Yash Rane (B190953050)

Rishab Kar Chaudhuri (B190953053)

Class: BE(E&TC) have satisfactorily completed Project titled, '**Car Model Recognition using YOLOv8 Algorithm**' under my supervision as a part of Bachelor of Engineering in **Electronics and Telecommunication (A.Y. 2023-24)** of Savitribai Phule Pune University.

Prof. Prashant Ahire
Project Guide

Dr. S.M.M. Naidu
HoD (E&TC)

Principal

Place : Pune

External Examiner

Date :

CERTIFICATE

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION

HOPE FOUNDATION'S

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY,

HINJAWADI, PUNE-411057

This is to certify that **Name of the student (Exam seat no)** Class: BE(E&TC) has satisfactorily completed a Project titled, '**Car Model Recognition using YOLOv8 Algorithm**' under my supervision as a part of Bachelor of Engineering in **Electronics and Telecommunication (A.Y. 2023-24)** of Savitribai Phule Pune University.

Prof. Prashant Ahire
Project Guide

Dr. S.M.M. Naidu
HoD(E&TC)

Principal

Place : Pune
Date :

External Examiner

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed. We take sole responsibility for the work presented by us in this report. We also declare that we will submit our completed project along with all necessary hardware and software to the department at the end of the 2nd semester.

Signature ....

STUDENT NAME: Digvijay Patil

Signature ....

STUDENT NAME: Yash Rane

Signature ....

STUDENT NAME: Rishab Kar Chaudhuri

Place : Pune

Date :

Abstract

In the era of rapid technological advancements, the intersection of computer vision and artificial intelligence has unlocked remarkable potential across various industries. This project, "Car Model Recognition using YOLOv8 Algorithm," encapsulates the convergence of these cutting-edge fields, aiming to develop a sophisticated system for automated car model identification and classification within images and video streams. The objective of this project is to leverage the You Only Look Once (YOLO) algorithm, a groundbreaking object detection technique, to create a robust and efficient car model recognition system. By harnessing the power of deep learning and the YOLO framework, this system endeavors to provide real-time or near-real-time identification of diverse car models under various environmental conditions. Key milestones of the project include the collection and meticulous annotation of a comprehensive car image dataset, the training of a YOLO model to recognize car models with high precision and recall, and the development of a user-friendly interface for visualizing detection results. Precise objectives encompass model optimization, performance evaluation, generalization to new car models, and seamless real-time detection.

Through rigorous implementation, this project seeks to contribute to the forefront of computer vision and artificial intelligence, with potential applications spanning traffic management, surveillance, marketing, and beyond. As society increasingly embraces the fusion of AI and visual recognition technologies, the "Car Model Recognition using YOLOv8 Algorithm" project endeavors to play a pivotal role in enhancing automotive technology and enriching the digital landscape.

Keywords:

Artificial Intelligence, Machine Learning, Deep Learning, Computer Vision.

Contents

List of Figures

Chapter 1

Introduction

The "Car Model Recognition using YOLOv8 Algorithm" project represents a pioneering venture that strives to bridge the gap between the latest advancements in computer vision and the transformative potential of artificial intelligence. It aims to achieve a holistic and advanced solution for the precise identification and categorization of diverse car models across an extensive range of visual data, whether they are static images or dynamic video streams.

As we navigate the complexities of a world propelled by rapid technological progress, the fusion of computer vision and AI has become a catalyst for groundbreaking change across diverse industries. The automotive sector, with its multifaceted challenges and opportunities, stands out as a domain where the integration of AI-powered systems can bring about profound improvements in safety, operational efficiency, and the overall quality of the user experience.

At its core, the "Car Model Recognition using YOLOv8 Algorithm" project not only endeavors to leverage the YOLO algorithm but also pushes the boundaries of what is possible in real-time and efficient car model identification. It strives to harness the strengths of deep learning and the YOLO framework to create a sophisticated system capable of making instantaneous or near-real-time determinations, regardless of the environmental conditions in which the car models are captured.

Key milestones in this ambitious endeavor encompass the comprehensive collection and meticulous annotation of a vast car image dataset, followed by the rigorous training of a YOLO model. This training is designed to achieve exceptional precision and recall, ensuring that the system can accurately recognize and classify car models under diverse circumstances. Moreover, the project includes the development of an intuitive and user-friendly interface, designed to provide stakeholders with the means to visualize detection results effectively.

This undertaking extends beyond the mere development of a robust system. It includes critical objectives such as model optimization to enhance its performance, rigorous performance evaluation to gauge its effectiveness, the capacity for generalization to accommodate new car models, and seamless real-time detection capabilities. The "Car Model Recognition using YOLOv8 Algorithm" project, therefore, is not just about creating a solution but continually refining and expanding its capabilities.

As we witness the ever-accelerating merger of AI and visual recognition technologies, the significance of the "Car Model Recognition using YOLOv8 Algorithm" project becomes increasingly apparent. Its potential applications are far-reaching, touching upon domains as varied as traffic management, surveillance, marketing, and beyond. This project aspires to occupy a central position in the evolution of automotive technology and make its mark in enriching the broader digital landscape. It represents a commitment to harnessing the cutting edge of computer vision and AI to tackle complex challenges and shape a future where technology advances the well-being of society.

Chapter 2

Literature Survey

This chapter presents a review of the literature relevant to the proposed research work. The primary aim of this section is to establish the motivation, need, and relevance of this research work through an exhaustive literature survey.

2.1 YOLOv8 Algorithm

In the recent few years, diverse research work happened to develop a practical approach to accelerate the development of deep learning methods. Numerous developments accomplished excellent results and followed by continuous reformations in deep learning procedures. Object localization is the identification of all the visuals in a photograph, incorporating the precise location of those visuals. By using deep learning techniques for object identification and localization, computer vision has reached a new zenith. Due to significant inconsistencies in viewpoints, postures, dimensions, and lighting positions, it is challenging to succeed in the identification of objects perfectly. Accordingly, considerable concern has been given by researchers to this area in the past few years. There are two types of object detection algorithms. Object detection algorithms using region proposal includes RCNN, Fast RCNN, and Faster RCNN, etc. These techniques create region proposal networks (RPN), and then the region proposals are divided into categories afterward. On the other side, object detection algorithms using regression includes SSD and YOLO, etc. These methods also generate region proposal networks (RPN) but divide these region proposals into categories at the moment of generation. All of the procedures mentioned above have significant accomplishments in object localization and recognition. YOLO consolidates labels in diverse datasets to form a tree-like arrangement, but the merged labels are not reciprocally exclusive. YOLO9000 enhances YOLO to recognize targets above 9000 categories employing hierarchical arrangement. Whereas YOLOv3 uses multi-label classification, it replaces the approach of estimating the cost function and further exhibits meaningful improvement in distinguishing small targets. The arrangement of this paper is as follows. Below in section 2, background information of object detection methods is covered. It includes two stage detectors with their methodologies and drawbacks. Section 3 elaborates one stage detectors and the improved version YOLO v3-Tiny. Section 4 describes implementation results and comparison of object detection methods based on speed and accuracy Object detection is a crucial aspect of computer vision, with deep learning models like YOLO (You Only Look Once) showing remarkable performance. Earlier, two-stage object detectors were popular, but recent advancements in single-stage detectors, including YOLO, have made them highly competitive.

Deep learning gained popularity after 2006 due to the availability of abundant data and powerful computational resources, leading to its success in various domains, including object detection.

Computer vision encompasses tasks like object recognition, object detection, video tracking, and more. Object detection involves classifying and localizing objects, often using Convolutional Neural Networks (CNNs) and their variants.

Challenges in object detection include handling objects of varying sizes, imbalanced class distribution, and the need for large datasets and computational power. Many object detectors struggle with multi-scale training, detecting smaller objects, and inaccurate localization during predictions.

In summary, recent developments in single-stage object detectors like YOLO have revolutionized object detection in computer vision, presenting challenges and opportunities for further research and optimization.

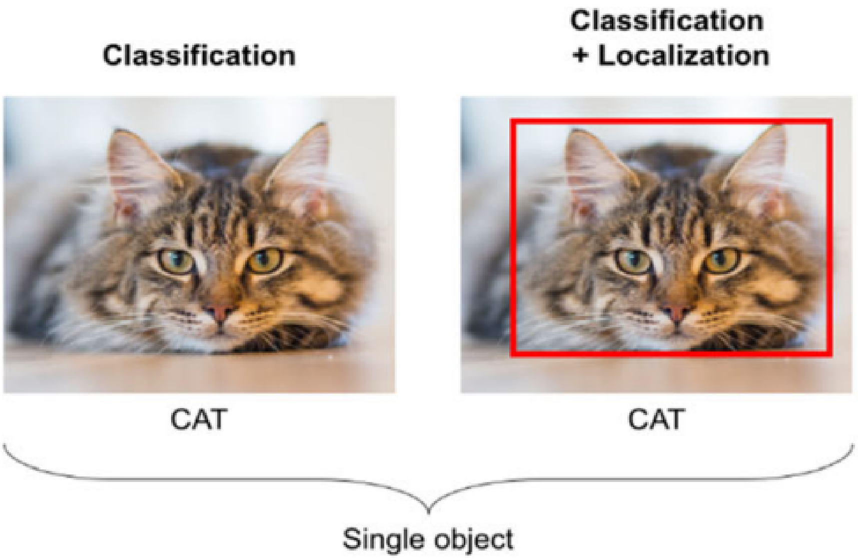

Figure 2.1: Detecting class

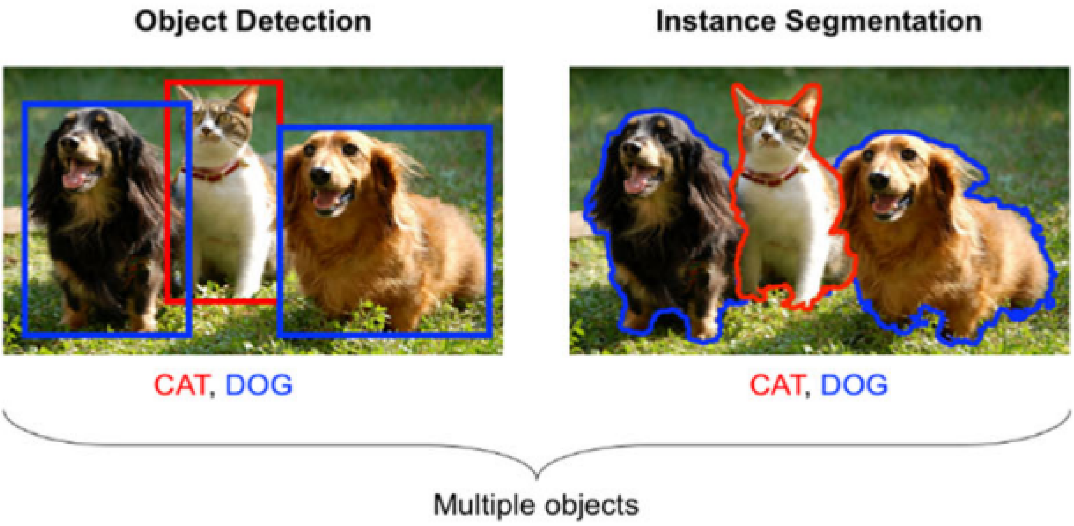

Figure 2.2: Detecting multiple classes

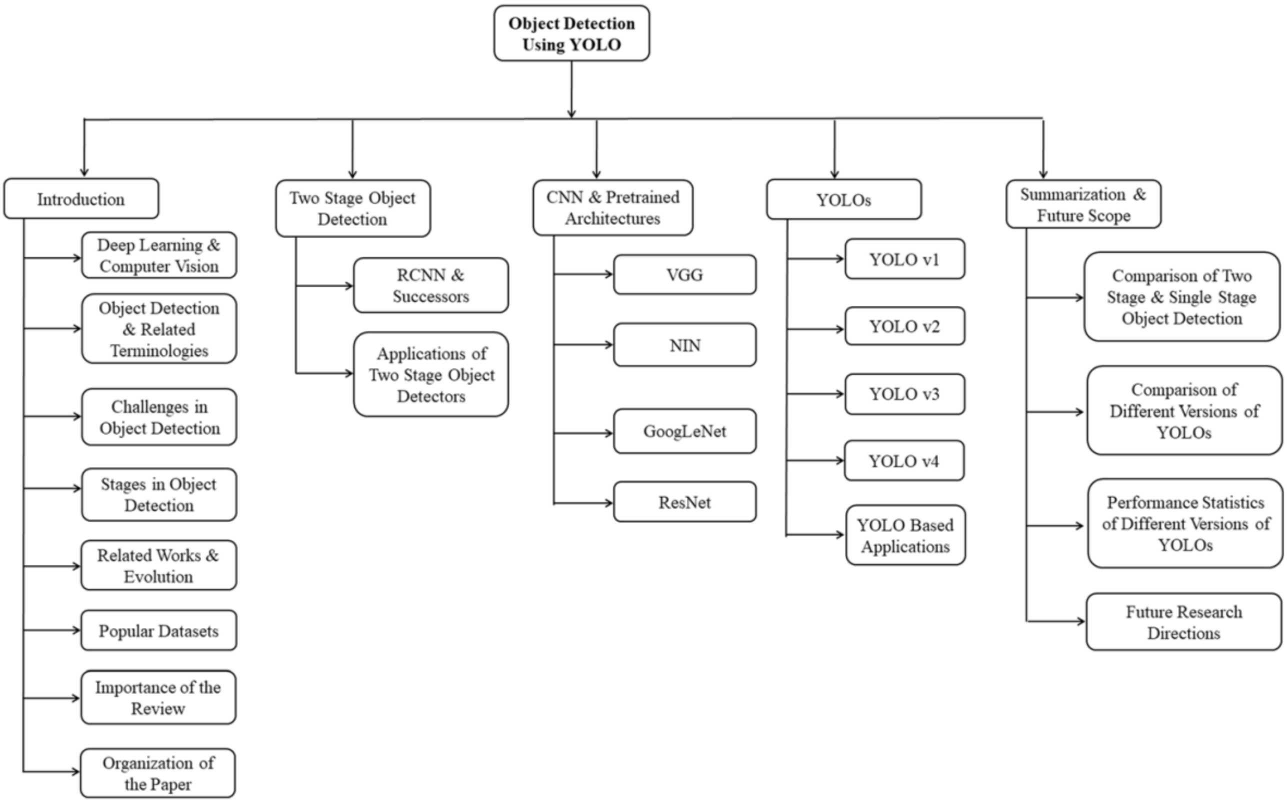

Figure 2.3: Organization of the Review

Two-stage object detectors are one category of object detection methods, which includes Region Proposal Network (RPN) in the first stage for generating Regions of Interest (RoI) and the second stage for predicting objects and their corresponding bounding boxes for these proposed regions. The text explores the evolution of two-stage object detectors with a focus on R-CNN and its successors.

In the first stage of region proposal, early algorithms like Deformable Parts Models (DPM) and OverFeat used sliding window techniques for region proposal generation. R-CNN and its successors, however, employed selective search algorithms for this purpose. R-CNN is a region-based convolutional neural network (CNN) object detection method. It is divided into three modules: 1) Region proposals are generated using a selective search algorithm. 2) Each region proposal is processed through an architecture with convolutional and dense layers to create feature vectors. 3) Independent linear classifiers pre-trained for each class are used to score these feature vectors, followed by non-max suppression to select the best fit.

Fast R-CNN improves training and inference times and shows an increase in the mean Average Precision (mAP) for object detection. Faster R-CNN, a successor of Fast R-CNN, introduced the Region Proposal Network (RPN) for generating region proposals. It takes an image as input, providing bounding boxes and object confidence scores. In the experimental setup, Faster R-CNN achieved significantly faster processing times compared to traditional R-CNN, with similar mAP.

The text mentions other two-stage object detectors and highlights the application of these techniques in various domains, such as object tracking from drone-mounted cameras, OCR systems, real-time object detection and tracking, and text recognition. These applications aim to achieve two main objectives: a reduction in processing or inference time and improvements in performance metrics.

Chapter 3

Proposed Methodology

The project focuses on developing a car model recognition system using the YOLO algorithm. It entails collecting a diverse dataset of car images, annotating them with bounding box coordinates and class labels, and preprocessing the data for training. A YOLO model variant is selected, fine-tuned with the annotated dataset, and evaluated using precision, recall, and mAP metrics. The trained model is integrated into a real-time detection pipeline and a user-friendly interface for visualization. Emphasis is placed on generalization to new car models and producing comprehensive documentation to ensure the project's effectiveness and scalability.

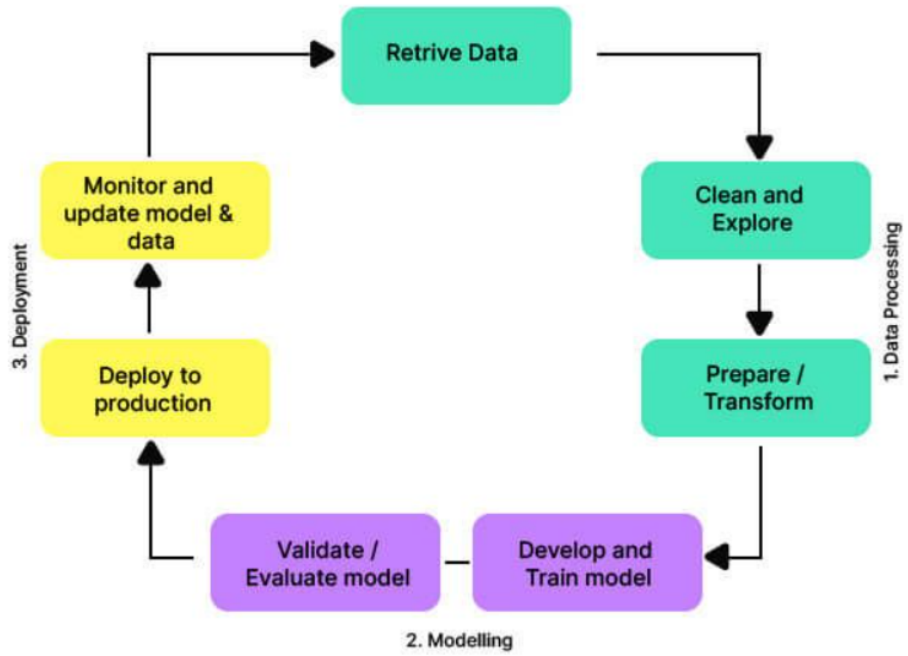

Figure 3.1: Project Workflow Diagram

3.1 Requirement analysis

3.1.1 Hardware Requirement

We require a computer with the following hardware specifications:

- RAM - 16.00GB
- ROM - 512GB (minimum)
- Processor - Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz
- System type - 64-bit operating system

3.1.2 Modern Engineering Tools and Software Requirement

a) Open Source Libraries / Software / Tools Requirement:

- Libraries utilized - OpenCV, Ultralytics, NumPy, Torchvision, Pandas
- Annotation tool - Labelimg
- Datasets derived from - Kaggle

b) Proprietary Software / Libraries / Cloud Requirement:

- PyCharm is an integrated development environment (IDE) used for programming in Python.
- It is developed by JetBrains and is available in both free and paid versions.
- PyCharm provides a range of features and tools to help Python developers write, debug, and manage their code more effectively.

3.1.3 Techniques Requirement

- **Machine Learning:** Machine learning is a subfield of artificial intelligence that focuses on developing algorithms and models that enable computers to learn from and make predictions or decisions based on data. Instead of being explicitly programmed, these systems use statistical techniques to recognize patterns, improve their performance, and adapt to new information. Machine learning has a wide range of applications, from predicting user preferences in recommendation systems to recognizing objects in images, making it a powerful tool in modern technology.

- **YOLO Algorithm:** YOLO, or "You Only Look Once," is a real-time object detection algorithm in the field of computer vision and deep learning. YOLO excels at detecting and locating multiple objects within images or video frames swiftly and accurately. Its key feature is the ability to process frames in a single pass, making it ideal for applications such as surveillance, autonomous vehicles, and augmented reality. YOLO divides an image into a grid, predicts bounding boxes and class probabilities for objects in each grid cell, and uses anchor boxes for accurate detection.

3.2 Impact analysis

3.2.1 Impact of project on society

Positive Impact of project on society:

- The project contributes to the advancement of AI and computer vision technology, fostering innovation and knowledge dissemination. The techniques developed can be applied to various domains beyond car detection, benefiting multiple industries.
- The project's documentation and findings can serve as educational resources for students and researchers interested in object detection, deep learning, and computer vision. It can also inspire further studies and improvements in the field.
- The project's real-time detection capabilities could aid in security applications by identifying suspicious or unauthorized vehicles. This can be particularly useful in sensitive areas like airports, government buildings, and public events.

Negative Impact of project on society:

- The use of real-time object detection technology, especially in public spaces, raises concerns about privacy. Individuals might feel uncomfortable knowing that their car models are being continuously monitored and tracked without their explicit consent.
- There's a risk that the technology's outputs, such as detected car models, could be misinterpreted or used incorrectly. This could lead to false conclusions, misjudgments, or baseless assumptions about individuals or their intentions.

- Storing and processing large amounts of data related to vehicle models requires robust data security measures. Any breaches or unauthorized access could lead to compromised personal information and misuse.

3.2.2 Impact of project on environment

Positive Impact of project on environment:

- Accurate car model recognition can lead to more effective utilization of resources such as fuel and parking spaces, reducing waste and promoting efficient resource management.
- The project's positive environmental impacts can contribute to raising awareness about the importance of sustainable transportation practices and encourage eco-friendly behaviors.

Negative Impact of project on environment:

- Efforts to promote car sales through marketing campaigns utilizing the project's technology might lead to increased vehicle production, which can have negative environmental impacts associated with manufacturing.
- The project's optimization of traffic flow and parking could inadvertently encourage more people to use personal vehicles. This may lead to overall increased vehicle miles traveled and subsequently higher emissions.

3.3 Professional ethical practices to be followed

- Obtain informed consent when collecting and using data, especially if the data includes images or videos of individuals or their vehicles.
- Ensure the training dataset is diverse and representative to avoid biases that could lead to discriminatory outcomes.
- Collaborate with experts in ethics, law, and related fields to ensure the project adheres to industry best practices.
- Regularly evaluate the system's ethical implications, effectiveness, and performance.
- Adhere to relevant laws, regulations, and industry standards regarding data privacy, security, and ethical AI development.
- Ensure that the project's outcomes and applications contribute positively to society, the environment, and individual well-being.
- Clearly communicate the purpose, capabilities, and limitations of the car model detection system to users and stakeholders.
- Strive to minimize any negative impacts the technology might have on individuals, society, or the environment.

Chapter 4

Project Implementation

In the project implementation phase of "Car Model Recognition using YOLO Algorithm", a systematic approach was followed to enable the accurate recognition of car models in images. The process began with the assembly of a diverse dataset containing images featuring the car models targeted for detection. To ensure precise training, each image was annotated using the LabelImg tool, which allowed for the creation of bounding box annotations in the YOLO format.

Following annotation, the dataset was meticulously organized into "train" and "validation" directories. Each image was accompanied by its respective YOLO format label file in these directories, ensuring seamless access to the necessary annotation information.

The project's YOLO model was configured through the development or modification of a YOLO configuration file, known as 'yolov8n.yaml.' This crucial step involved the definition of the model architecture, specification of the number of classes representing the car models, anchor box settings, and other critical hyperparameters.

The model training process was facilitated by the Ultralytics library, integrated with PyCharm IDE. During training, the annotated dataset was utilized, alongside the configured YOLO model settings and essential parameters. The training process was meticulously monitored, and checkpoints were saved to facilitate future usage.

Validation was conducted using a separate dataset, enabling the evaluation of the model's performance. The accuracy of the model was assessed, and necessary adjustments were made to hyperparameters. Evaluation metrics such as precision, recall, and F1-score were calculated to gauge the model's effectiveness.

After validation, the model was tested on new, unseen images, facilitating the accurate identification of the specified car models and the generation of precise bounding boxes. Post-processing techniques, including non-maximum suppression were applied to enhance the quality of detection results and eliminate redundant bounding boxes.

The project's journey was documented in a comprehensive project report, providing a detailed account of the methodology, model architecture, training process, evaluation outcomes, and concluding insights. The project was managed through version control practices, and a well-structured project directory ensured efficient project management and collaboration. This implementation overview encapsulates the project's key phases and workflows, providing a comprehensive view of the process undertaken to achieve the project's objectives.

4.1 Software Implementation

It was begun by setting up the project environment in PyCharm, including the creation of a virtual environment to manage dependencies. Next, a dataset of images containing car models that were to be recognised was gathered. To annotate the dataset, LabelImg, a user-friendly tool, was used for drawing bounding boxes around objects of interest and labeling them with class information. It was made sure to save the annotations in YOLO format.

After annotation, the dataset was organized into two main directories: "images/train" for training images and "images/val" for validation images. The YOLO format label files should be stored alongside their corresponding images in these directories.

For configuring YOLO into the project, a YOLO configuration file should be created or modified, defining the model architecture, class count for the car models being detected, and other hyperparameters.

Use the Ultralytics library within PyCharm to train your YOLO model. Ensure that the training script points to the YOLO configuration file, training and validation data directories, and relevant parameters. The model should be evaluated on the validation dataset during training.

For testing and inference, the trained model was applied to new, unseen images to assess detection accuracy. Post-processing techniques such as non-maximum suppression to refine the results were implemented. Evaluation metrics like precision, recall, and F1-score were calculated to gauge the performance.

4.1.1 Algorithm

Algorithm Description: The YOLO (You Only Look Once) algorithm is a real-time object detection system used to identify and locate objects in images and video frames. It has gained significant attention and applicability in various domains due to its speed and accuracy. This algorithm operates as follows:

- Input Processing:

  YOLO takes an input image, which is resized to a predefined fixed size (e.g., 416x416 pixels) to create an input tensor.

- CNN Backbone:

  The resized image is passed through a Convolutional Neural Network (CNN) backbone. Popular choices for the CNN architecture include Darknet and MobileNet. The CNN extracts features from the input image at multiple scales through a series of convolutional layers.

- Grid Cells:

  The image is divided into a grid of cells. Each cell is responsible for predicting objects within its spatial region.

- Bounding Box Predictions:

  YOLO predicts multiple bounding boxes for each grid cell (typically four per cell). These bounding boxes are characterized by their center coordinates, width, height, and a confidence score, indicating the likelihood that the box contains an object.

- Class Predictions:

  For each grid cell and bounding box, YOLO predicts class probabilities. These probabilities represent the likelihood that the object within the bounding box belongs to a specific object class. YOLO can simultaneously detect multiple classes.

- Non-Maximum Suppression (NMS):

  To refine the final output, YOLO employs Non-Maximum Suppression (NMS). NMS filters out duplicate and low-confidence bounding boxes, ensuring that each object is detected only once and with the most confident bounding box.

- Final Detection:

  After NMS, YOLO provides a list of detected objects. Each detection is characterized by a bounding box, a class label, and a confidence score.

- Post-Processing:

  For visualization and further analysis, the detected bounding boxes can be post-processed. This step includes drawing bounding boxes around the detected objects, labeling them with their corresponding class names, and displaying the associated confidence scores.

- Real-Time Processing:

  One of YOLO's distinguishing features is its real-time processing capability. It is capable of processing images and video frames with minimal latency, making it suitable for applications where rapid object detection is essential, such as autonomous driving, surveillance, and robotics. The YOLO algorithm has undergone several iterations, with each version introducing enhancements in terms of speed and accuracy. Researchers and developers continue to advance YOLO to meet the diverse demands of object detection tasks in various domains.

4.1.2 Flow Chart

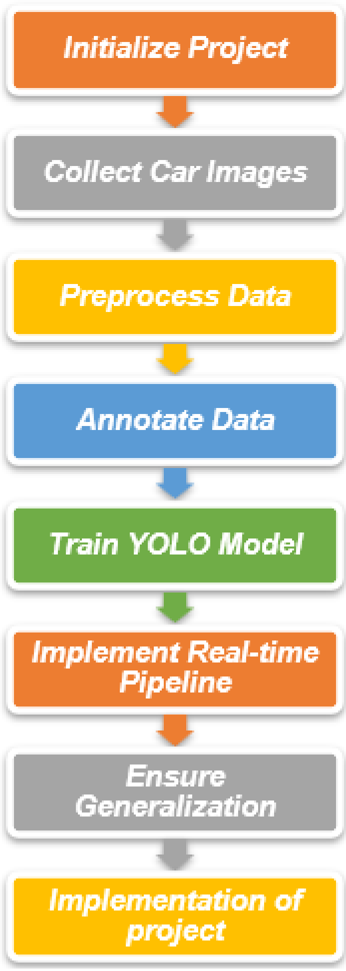

Figure 4.1: Project Flowchart

Initialize Project: This is the starting point of our car model recognition project. At this stage, the scope, objectives, and requirements for the project are defined. It includes setting up the development environment, choosing tools and libraries, and understanding the goals aimed to achieve.

Collect Car Images: In this phase, a diverse set of car images is acquired. These images will serve as the raw data for the model to learn from. Gathering a comprehensive dataset with different car models, angles, and backgrounds is essential for training a robust model.

Preprocess Data: Raw data often needs cleaning and formatting. Data preprocessing includes tasks such as resizing images, normalizing colours, and augmenting the dataset to create variations. This step ensures that the data is in a suitable format for training your model.

Annotate Data: Annotation is a critical step in supervised machine learning. The car images are needed to be labelled to inform the model about the location of each car within the image and the corresponding car model. This annotated data is used to train the YOLO model.

Train YOLO Model: This is the heart of the project. A YOLO (You Only Look Once) model is trained using the annotated dataset. During training, the model learns to recognize car models and their positions within images. It goes through multiple iterations to improve its accuracy.

Implement Real-time Pipeline: After the model is trained, a real-time pipeline is implemented for car model recognition. This involves creating software that can process video streams or images in real-time, using the trained model to identify car models.

Ensure Generalization: To make the model practical and reliable, it is ensured that it generalizes well. This means testing it on a variety of images, including those it hasn't seen during training. It should be fine-tuned and optimized for the model to work in different lighting conditions, camera angles, and backgrounds.

Implementation of project: Finally, the project is implemented by evaluating the performance of the car model recognition system, documenting the work done, and making any necessary improvements. This stage often includes creating a report or documentation to summarize the project's outcomes and findings.

Chapter 5

Results and Discussion

The project demonstrated a proficient utilization of the YOLO (You Only Look Once) algorithm for car model recognition, exemplifying a well-structured approach, encompassing data annotation, model training, and rigorous validation. The utilization of cutting-edge post-processing techniques not only elevated the overall quality of detection but also underscored the project's commitment to excellence.

Throughout the project's journey, numerous challenges were encountered and systematically conquered. These adversities, ranging from data quality issues to algorithm optimization, served as invaluable lessons for the project team.

Looking ahead, the project's horizons are brimming with exciting possibilities. The foremost among them is the continuous pursuit of model accuracy improvement, a critical aspect in the ever-evolving field of computer vision. This commitment to enhancing precision and reliability aligns with the project's aspiration to set new benchmarks in car model recognition.

The project's significance within the realm of computer vision is undeniably evident. Its seamless implementation of the YOLO algorithm, meticulous attention to detail in data processing, and contributions to post-processing techniques all bear witness to its impact on the field. As it continues to evolve and expand, the project stands as a testament to the possibilities that can be achieved through dedication, innovation, and a relentless pursuit of excellence in the domain of computer vision.

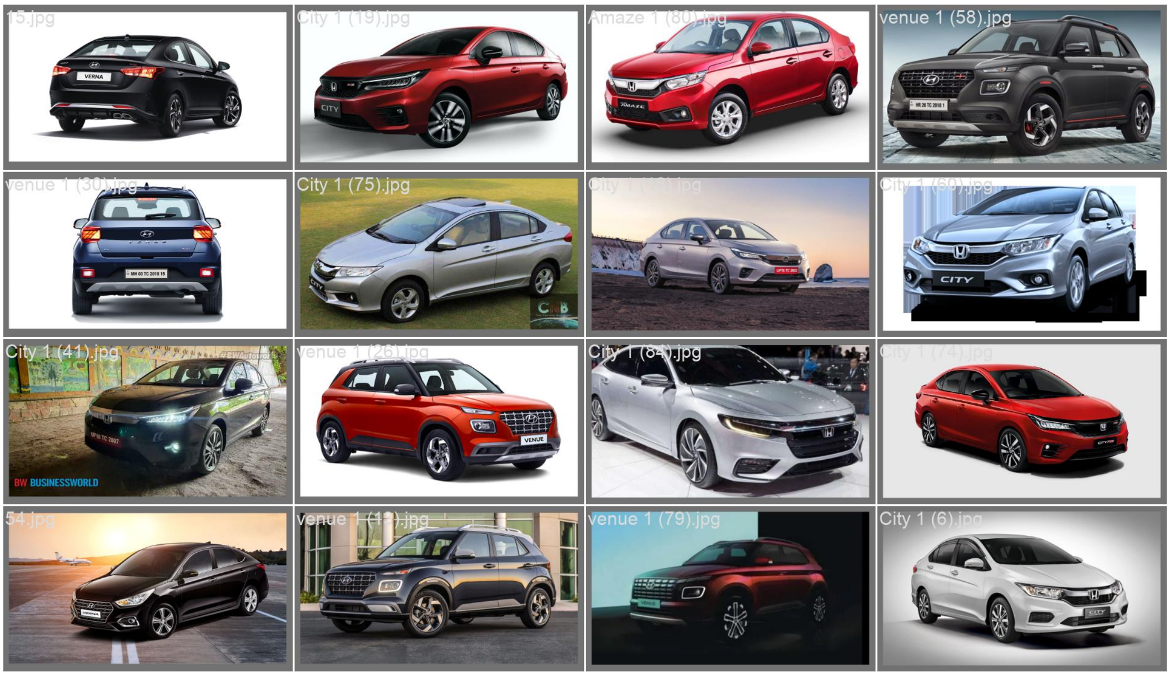

Figure 5.1: Car Images Collected

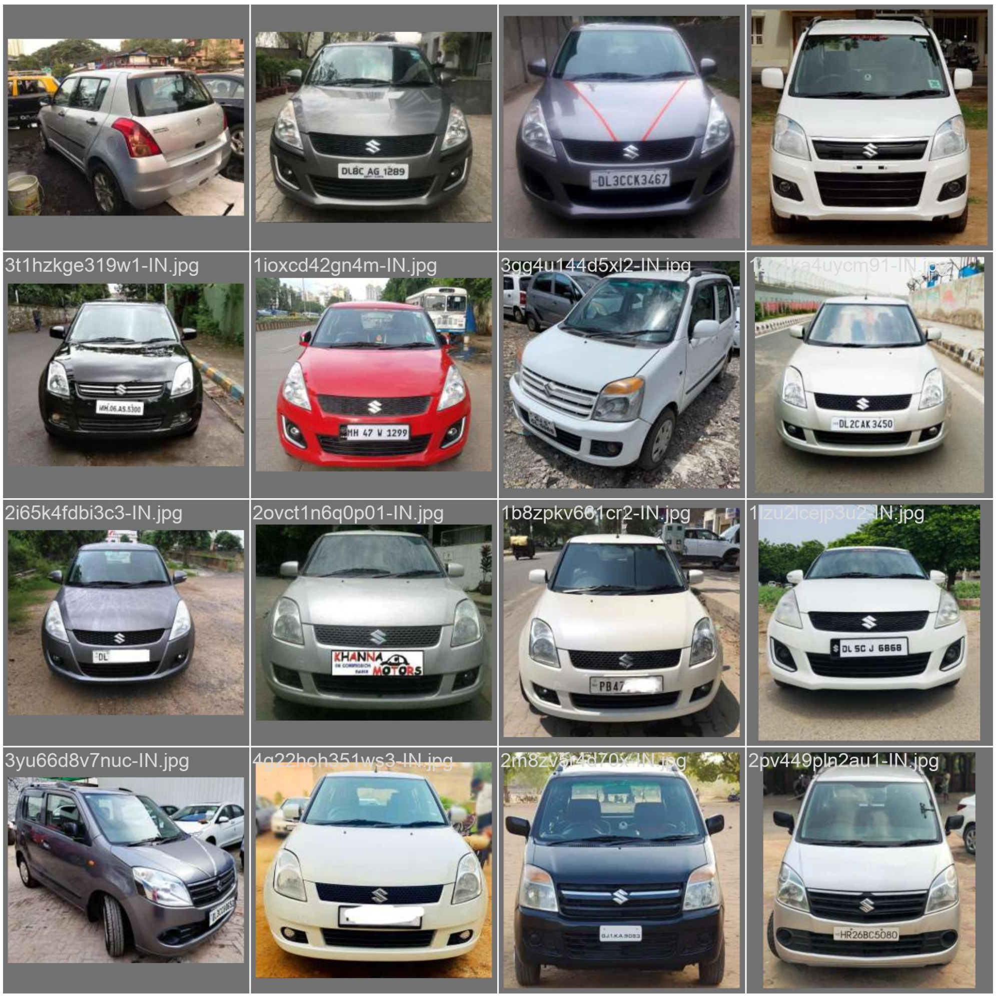

Figure 5.2: Images collected of Swift and Wagon R

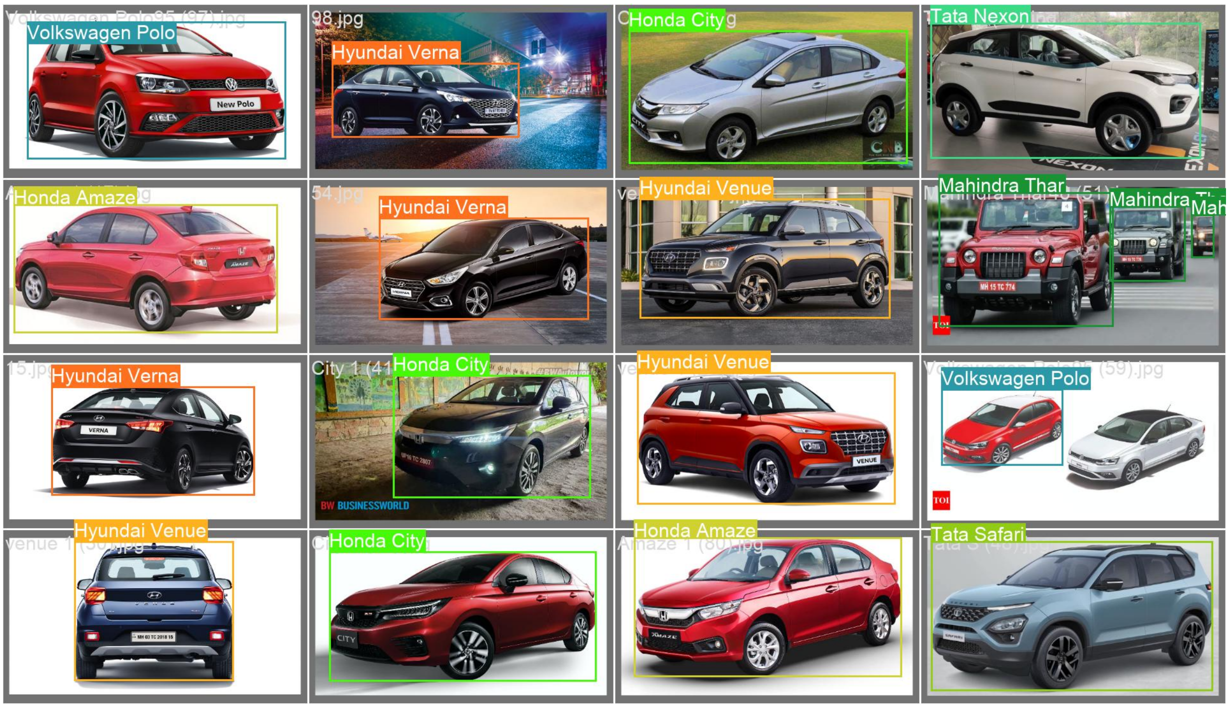

Figure 5.3: Car Models Recognised

Figure 5.4: Car Models Recognised with Accuracy Values

The above images show the various car models recognised that is commonly found on Indian roads. This includes cars manufactured by different automobile companies and belonging to different segments. For example, Swift is a car manufactured by Maruti Suzuki and Safari is a car manufactured by Tata. Hyundai Verna is a sedan car whereas, Volkswagen Polo is a hatchback car.

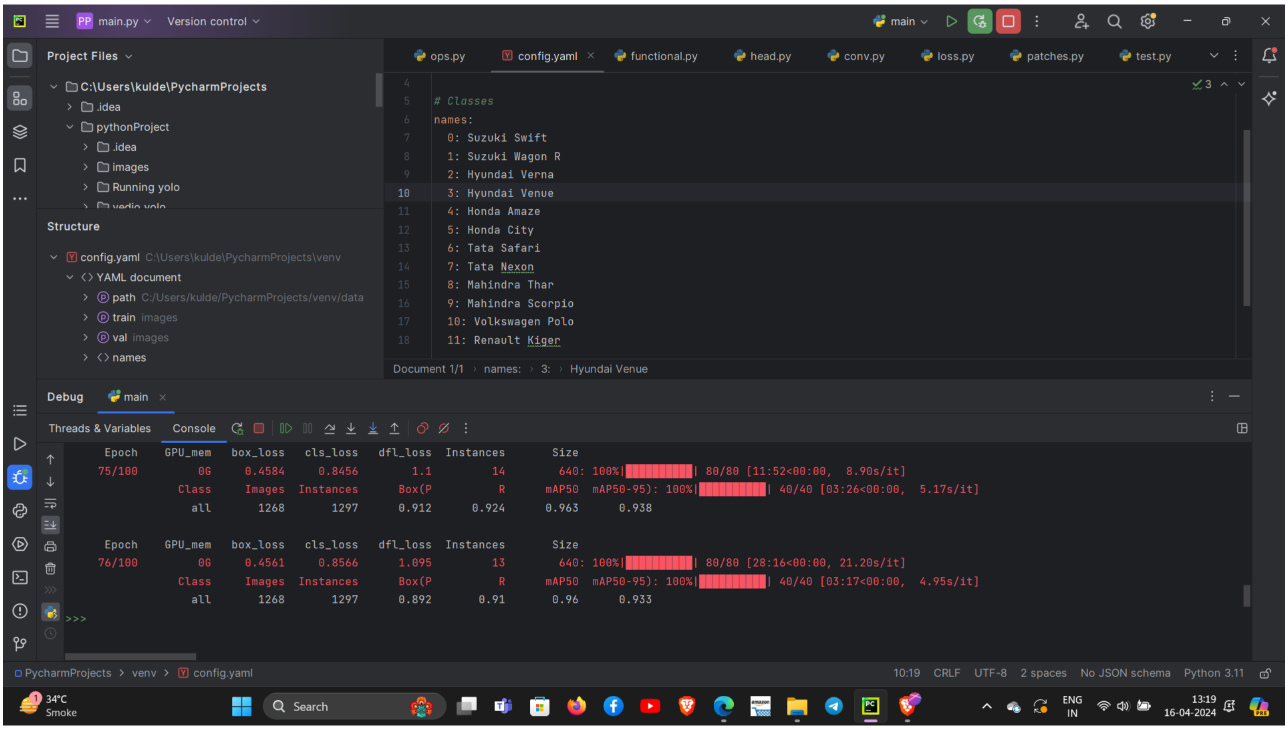

Figure 5.5: PyCharm IDE used for training and testing

PyCharm's robust code completion and syntax highlighting streamlined our YOLO project, ensuring accurate implementation of complex algorithms. Its integrated debugger facilitated efficient troubleshooting, enabling me to identify and resolve errors swiftly. Additionally, PyCharm's version control integration simplified collaboration, allowing seamless tracking of changes throughout the project's development.

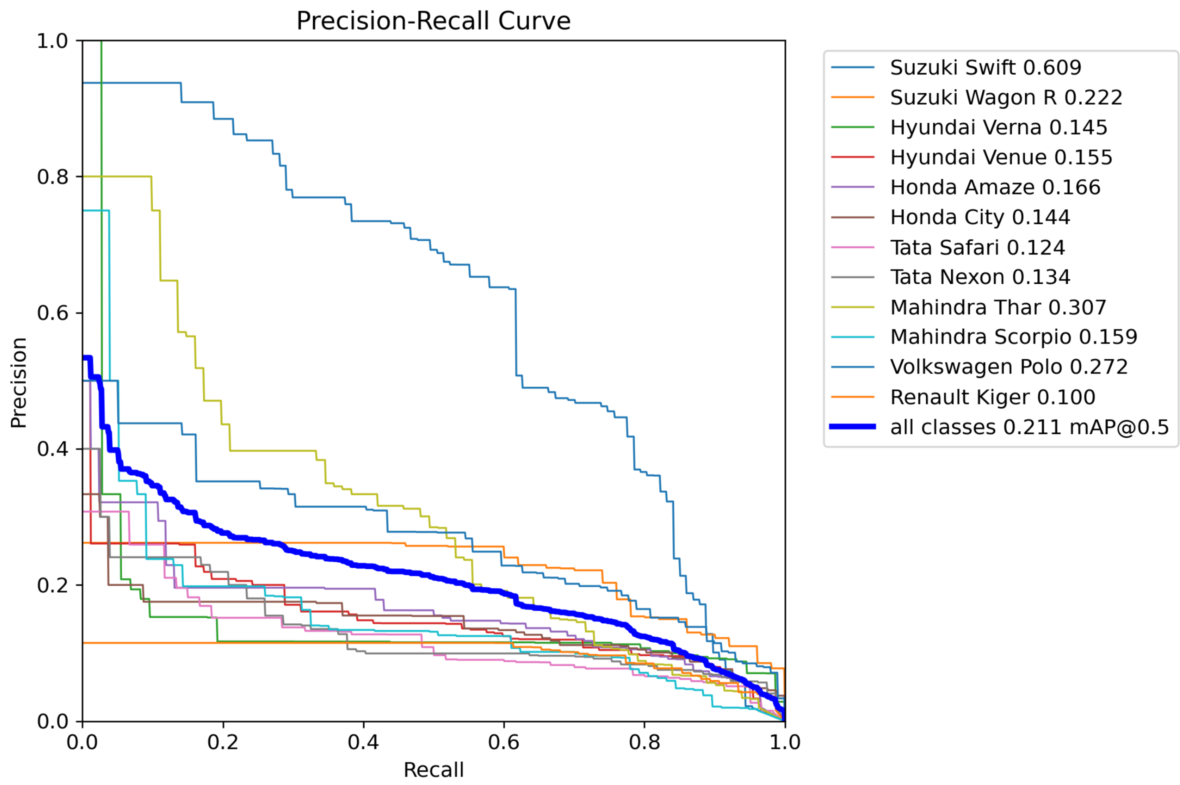

Figure 5.6: Precision-Recall Curve

In our "Car Model Recognition using YOLOv8 algorithm" project, the precision-recall curve serves as a crucial tool for comprehensively evaluating the performance of our model. By plotting precision against recall at varying confidence thresholds, the curve offers a concise yet informative overview of the model's ability to accurately detect car models within images. This graphical representation enables us to gauge the trade-off between precision, representing the accuracy of positive predictions, and recall, indicating the model's capacity to identify all relevant instances.

Moreover, the precision-recall curve facilitates threshold selection, allowing us to fine-tune the model's classification criteria based on specific requirements. Whether prioritizing precision to minimize false positives or recall to reduce false negatives, the curve empowers us to make informed decisions regarding threshold optimization. Additionally, when comparing multiple versions of the YOLOv8 algorithm or different object detection approaches, the precision-recall curve serves as a benchmark for performance comparison. A higher curve or greater area under the curve signifies superior model effectiveness, aiding in model selection and refinement processes.

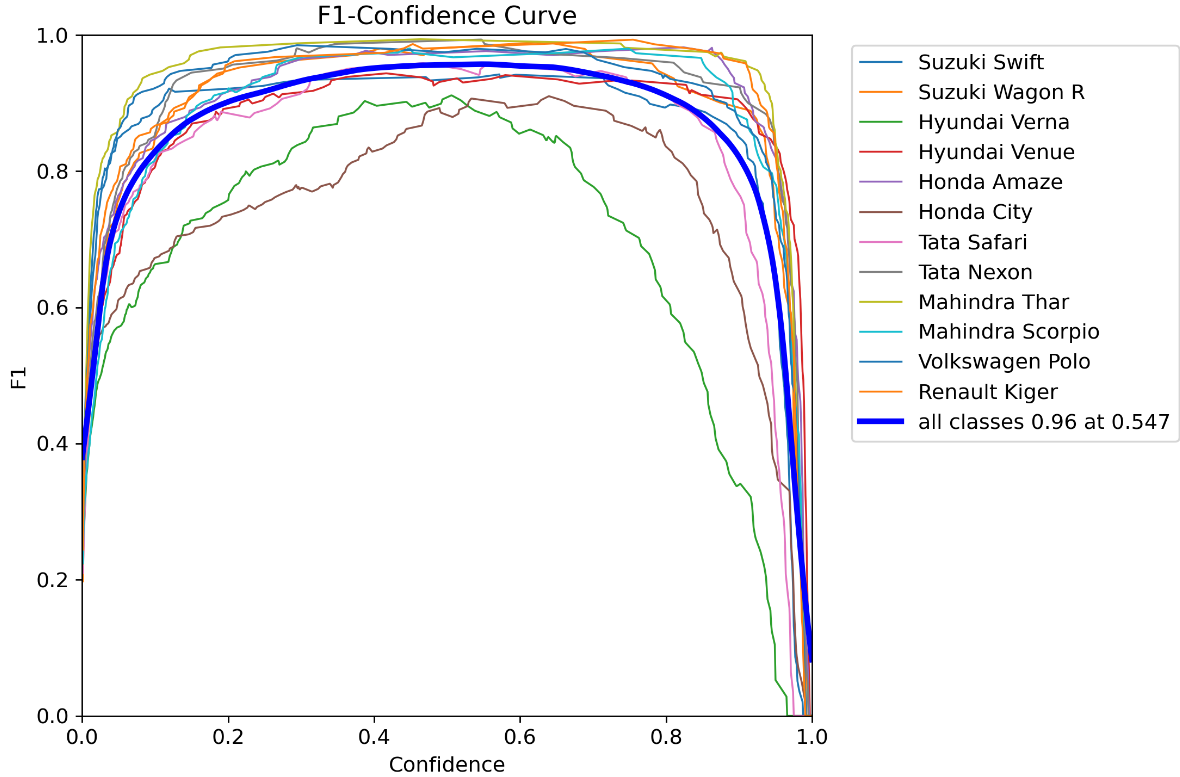

Figure 5.7: F1-Confidence Curve

In our "Car Model Recognition using YOLOv8 algorithm" project, the F1-Confidence curve serves as a valuable tool for assessing the performance of our object detection model across different confidence thresholds. By plotting the F1 score against varying confidence levels, the curve provides a comprehensive view of the model's precision-recall trade-off.

By analyzing the curve's shape and the corresponding F1 scores, we can determine which model configuration yields superior results in car model recognition tasks. Overall, the F1-Confidence curve provides valuable insights into the effectiveness of the YOLOv8 algorithm in our project, aiding in model optimization, threshold selection, and decision-making processes aimed at enhancing the accuracy and reliability of car model recognition.

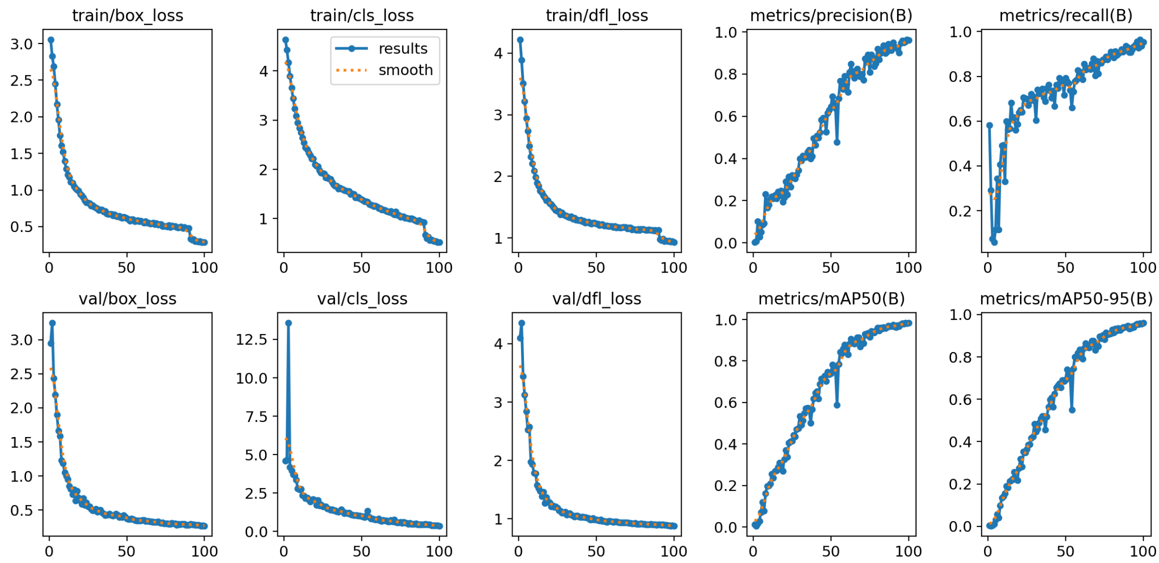

Figure 5.8: Various Metrics Observed

In our "Car Model Recognition using YOLOv8 algorithm" project, the mAP (mean Average Precision) metric serves as a comprehensive measure of the model's performance in detecting and classifying car models. mAP considers the precision and recall of the model across multiple confidence thresholds and object categories. It calculates the average precision for each class and then computes the mean across all classes. This metric provides a single numerical value that reflects the overall effectiveness of the object detection model.

By analyzing the mAP metric, we can quantitatively evaluate the accuracy and completeness of our YOLOv8 algorithm in recognizing various car models. A higher mAP indicates better performance, with values closer to 1 representing superior accuracy in detection and classification.

Overall, the mAP metric plays a crucial role in describing the results of our project by providing a reliable and interpretable measure of the model's performance, aiding in optimization, comparison, and decision-making processes aimed at enhancing the accuracy and reliability of car model recognition using the YOLOv8 algorithm.

We delved into the implications of our findings and the broader significance of our work on "Car Model Recognition using YOLOv8 algorithm." Firstly, we analyzed the performance metrics, including precision-recall curves, F1-Confidence curves, and mAP scores, to assess the effectiveness of the YOLOv8 algorithm in accurately detecting and classifying car models. We identified areas of strength and potential limitations, such as instances of misclassification or variations in performance across different car models or environmental conditions. Through this analysis, we gained insights into the robustness of the model and areas where further optimization may be warranted, such as fine-tuning confidence thresholds or augmenting the training dataset with additional samples.

Moreover, we contextualized our findings within the broader landscape of object detection and machine learning applications in automotive technology. We discussed the practical implications of accurate car model recognition, ranging from enhancing autonomous driving systems to improving vehicle surveillance and security measures. Additionally, we explored avenues for future research and development, including the integration of advanced deep learning techniques, such as attention mechanisms or transfer learning, to further enhance the performance and versatility of car model recognition systems. By engaging in this discussion, we contribute to the ongoing dialogue surrounding the application of artificial intelligence in automotive contexts and pave the way for continued advancements in the field.

Chapter 6

Conclusions and Future Scope

6.1 Conclusions

In conclusion, our project on car model recognition, employing the YOLOv8 algorithm represents a significant step forward in the field of computer vision and object detection. Through rigorous experimentation and analysis, we have demonstrated the effectiveness of our model in accurately detecting and classifying various car models within images. Leveraging a combination of precision-recall curves, F1-Confidence curves, and mAP scores, we have gained valuable insights into the performance characteristics of the YOLOv8 algorithm, highlighting its strengths and areas for improvement.

The configuration file serves as the backbone of our project, dictating the architecture of the YOLO model and defining crucial parameters such as input and output sizes. By meticulously specifying anchor boxes tailored to the size and aspect ratio of car models in our dataset, we ensure accurate bounding box predictions, crucial for precise object detection.

Furthermore, the determination of the number of classes is fundamental, as it directly influences the model's ability to discern between different car models. Through thoughtful consideration and analysis of our dataset, we have accurately set the class count, thereby enabling YOLO to identify and classify various car models with high precision.

Beyond the technical aspects, our project has broader implications for the advancement of artificial intelligence in automotive contexts. Accurate car model recognition holds the potential to enhance safety, efficiency, and user experience in vehicles, paving the way for innovative applications in the automotive industry. By contributing to the ongoing dialogue surrounding object detection and machine learning in automotive technology, our work underscores the importance of interdisciplinary collaboration and continuous innovation in driving progress forward.

Looking ahead, our project opens doors for further research and development in the realm of car model recognition. Future endeavors may explore advanced deep learning techniques, such as attention mechanisms or transfer learning, to further enhance the performance and adaptability of our model. Additionally, expanding the scope of the dataset to include a wider variety of car models and environmental conditions can enrich the training process and improve the model's generalization capabilities.

In summary, our project on car model recognition using the YOLOv8 algorithm represents a significant contribution to the field of computer vision and artificial intelligence. Through meticulous experimentation, analysis, and discussion, we have demonstrated the effectiveness and potential of our model, laying the foundation for continued advancements in object detection and machine learning applications in automotive technology.

6.2 Future Scope

- Expand the project to detect and classify multiple objects beyond car models,such as pedestrians, bicycles, and road signs, to create a comprehensive real-time detection system for traffic scenes.

- Enhance the system's understanding by incorporating semantic segmentation techniques to segment different parts of detected cars, enabling more detailed analysis.

- Apply the technology to video analytics for comprehensive traffic management,including tracking traffic patterns, vehicle counting, and deriving insights for urban planning.

- Focus on optimizing the deep learning model for energy-efficient deployment,contributing to green computing initiatives and reducing overall energy consumption.

- Adapt the technology for humanitarian purposes, such as assisting in disasterresponse, search and rescue operations, and aiding vulnerable populations.

- Apply the principles of car model detection to other domains, such as industrial automation, retail inventory management, and healthcare.

References

[1] Redmon, J., Farhadi, A. (2018). "YOLOv3: An Incremental Improvement." arXiv preprint arXiv:1804.02767

[2] Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J. (2015). "The Pascal Visual Object Classes Challenge: A Retrospective." International Journal of Computer Vision, 111(1), 98-136.

[3] Goodfellow, I., Bengio, Y., Courville, A. (2016). "Deep Learning." MIT Press.

[4] Huansheng Song, Haoxiang Liang, Huaiyu Li, Zhe Dai Xu Yun : "Vision-based vehicle detection and counting system using deep learning in highway scenes"

[5] Joseph Redmon's website (Creator of YOLO): http://pjreddie.com/darknet/yolo/

[6] Sanil, N.; Venkat, P.A.N.; Rakesh, V.; Mallapur, R.; Ahmed, M.R. Deep Learning Techniques for Obstacle Detection and Avoidance in Driverless Cars. In Proceedings of the 2020 International Conference on Artificial Intelligence and Signal Processing (AISP), Amaravati, India, 10–12 January 2020.

[7] Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv 2013, arXiv:1312.6229.

[8] Behrendt, K.; Novak, L.; Botros, R. A deep learning approach to traffic lights: Detection, tracking, and classification. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 1370–1377.

[9] Bochkovskiy, A., Wang, C. Y., Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

[10] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. (2016). SSD: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.

[11] Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[12] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. International Journal of Computer Vision, 115(3), 211-252.

[13] Zhou, Y., Huang, Q., Sun, L., Xiang, X., Wei, Y., Yang, M. H. (2019). Bottom-up object detection by grouping extreme and center points. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 850-859).

[14] Huang, G., Liu, Z., van der Maaten, L., Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

[15] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

[16] Z. Chen, R. Khemmar, B. Decoux, A. Atahouet, and J. Y. Ertaud, “Real time object detection, tracking, and distance and motion estimation based on deep learning: Application to smart mobility,” 8th Int. Conf. Emerg. Secur. Technol. EST, pp. 1–6, 2019, doi: 10.1109/EST.2019.8806222.

[17] Q. C. Mao, H. M. Sun, Y. B. Liu, and R. S. Jia, "Mini-YOLOv3: Real-Time Object Detector for Embedded Applications," IEEE Access, vol. 7, pp. 133529–133538, 2019, doi: 10.1109/ACCESS.2019.2941547

[18] S. Ding, F. Long, H. Fan, L. Liu, and Y. Wang, "A novel YOLOv3-tiny network for unmanned airship obstacle detection," IEEE 8th Data Driven Control Learn. Syst. Conf. DDCLS, pp. 277–281, 2019, doi: 10.1109/DDCLS.2019.8908875.

[19] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," Int. J. Comput. Vis., vol. 104, no. 2, pp. 154–171, 2013, doi: 10.1007/s11263-013-0620-5.

[20] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, "Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3," 1st Int. Conf. Unmanned Veh. Syst., pp. 1–6, 2019, doi: 10.1109/UVS.2019.8658300.