Vehicle Model Detection

A major project report submitted in partial fulfillment of the requirement for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering / Information Technology

Submitted by

Vaidehi Pandey(201156)

Kaushik Sharma(201149)

Under the guidance & supervision of

Dr. Kapil Rana

Department of Computer Science & Engineering and Information Technology

CERTIFICATE

This is to certify that the work which is being presented in the project report titled “Vehicle Model Detection” in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering and submitted to the Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat, is an authentic record of work carried out by “Vaidehi Pandey, 201156” and

“Kaushik Sharma, 201149” during the period from August 2023 to May 2024 under the supervision of Dr. Kapil Rana, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

Vaidehi Pandey	Kaushik Sharma
Roll No.: 201156	Roll No.: 201149

The above statement is correct to the best of my knowledge.

Jaypee University of Information Technology, Waknaghat, Solan - 173234 (India)

Candidate's Declaration

I hereby declare that the work presented in this report entitled **'Vehicle Model Detection'** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering / Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2023 to May 2024 under the supervision of **Dr. Kapil Rana** (Assistant Professor, Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature with Date)

Student Name: Kaushik Sharma	Vaidehi Pandey
Roll No.: 201149	201156

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature with Date)
Supervisor Name: Dr.Kapil Rana
Designation: Assistant Professor
Department: Computer Science and Engineering

ACKNOWLEDGEMENT

I would like to extend my heartfelt gratitude to all those who have played a crucial role in the successful completion of this major project. This endeavor has been a culmination of dedication, collaboration, and relentless effort from a diverse group of individuals, each contributing their unique skills and perspectives.

First and foremost, I express my deepest appreciation to my project supervisor Dr.Kapil Rana, whose guidance and expertise have been invaluable throughout every stage of this project. Their mentorship provided the necessary direction and insights, shaping the project into its final form.

I extend my appreciation to Jaypee University of Information Technology for providing the necessary resources and a conducive environment for the project`s execution.

Finally, my sincere thanks go to my family and friends for their unwavering support, understanding, and patience during the challenging phases of this project. Their encouragement provided the emotional sustenance needed to navigate the complexities of such a substantial undertaking.

In conclusion, the successful completion of this major project stands as a testament to the collective effort and dedication of all those mentioned above. It is with deep appreciation that I acknowledge their contributions, and I look forward to the continued success and collaboration in future endeavors.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Research on reliable and precise techniques for vehicle model detection has become essential due to the expanding use of surveillance systems and the requirement for effective traffic control. The goal of this project is to use computer vision techniques to automatically recognise and classify vehicle models from video and image data.

The project's main elements are feature extraction, data preparation methods, and model training with annotated datasets. Potential uses for the created system include urban planning, police enforcement, and traffic monitoring. It provides a scalable and affordable way to automate the identification of vehicle models, improving the effectiveness of traffic control and surveillance systems.

The research goes deeper into the particulars of data preprocessing, where methods like image scaling, augmentation, and normalization are used to make sure the model is resilient and can be used in a variety of environmental settings. In order to improve the system's capacity to distinguish minute variations across different models, significant spatial and temporal information are extracted from the identified cars throughout the feature extraction procedure. To balance accuracy and computing efficiency and maximize overall performance, a thorough examination of hyperparameters and model topologies is carried out..

The project's results highlight the significance of precise and dependable vehicle model identification for several social and industrial applications, making a significant contribution to the larger field of computer vision and deep learning applications in real-world scenarios.
The project report also discusses the suggested system's scalability and resource needs in light of its implementation in large-scale, real-world scenarios. Enhancements for edge computing

and real-time processing aspects are investigated to guarantee the system's viability in contexts with limited resources.

The proposed system's real-world applications are explored, including its possible contribution to smart city projects, traffic flow optimisation, and security enhancement through intelligent surveillance. The project's contributions to the more general domains of computer vision and deep learning are considered in the report's conclusion, along with insights into the difficulties encountered, the lessons discovered, and potential future directions for study and development in the dynamic field of automated vehicle model detection.

In conclusion, the project report offers a thorough analysis of the creation, difficulties, moral issues, and potential applications of a car model detection system. It also makes a significant contribution to the domains of deep learning, computer vision, and the responsible use of AI in traffic control and surveillance.

CHAPTER1: INTRODUCTION

1.1 INTRODUCTION

The conjunction between the computer vision and automotive era during this generation has provided novel usages. Vehicles' identification and classification has become an important piece of virtually all fields such as traffic control, safety, and other realms. This project takes advantage of recent developments such as deep learning and CNNs, using MobileNetV2 architecture to build a rigorous model for detecting vehicles.

This detector uses a lightweight vision model, namely MobileNetV2 which is very efficient and precise in mobile and embedded image recognition situations. This specific project is able to correctly discern car model types through image recognition or a live video feed by tapping into the vast energy that is embedded within it.

The core objectives of this project include:
Model Training

The use of mobileNetV2 model on a dataset made up of different car models in order to pick up fine details that are specific for every car type.

Real-time Detection: An effective pipeline for real-time vehicle model detection applicable for situations like traffic monitoring, parking management and security surveillance.

Accuracy and Adaptability: Classification with high accuracy that is also flexible enough for different levels of light intensities, angles, and other real environmental challenges.

This project aims to demonstrate how to implement MobileNetV2 vehicle model detector into computer's world. Ultimately, it must be able to offer a dependable as well as scalable design which can help support car detection functions for modern-day intelligent transport systems.

Within the last couple of years, deep learning and computer vision have become major contributors to the improvement of object detection accuracy and efficiency for various tasks, including the detection of vehicles type. One major aspect to such models success is precisely the application of data augmentation techniques that are aimed at enriching the dataset variety and complexity. On top of that, the algorithms of color detection are also very important for the recognition and differentiation of cars by their color characteristics. In this part, we give initial orientation on the augmentation techniques and color detection methods in the framework of vehicle model detection, highlighting why they are so crucial and how they influence the models performance.

Data augmentation is a technique which helps to increase the variety of training data by applying different transformations and modifications to the original images. In this instance, the persistent role of augmentation is to contribute to the stability and applicability of deep learning algorithms in the detection of vehicle models. Through perturbing in the training data, the model becomes capable to identify similar stances of vehicle taken from various viewpoints, lighting and occlusion conditions.

Common augmentation techniques used in vehicle model detection include:

Rotation: Such maneuvers as rotation of the vehicle images in different angles and positions, by certain degree, can serve to provide the impression that one is looking at the real thing.

Scaling: The resizing of the vehicle images to varying scales to simulate the variations in distance and size.

Horizontal and Vertical Flipping: By doing a horizontal or vertical reflections of photos , the effect of reflection or mirror reproduction is rescued .

Translation: Using the vehicle visuals and toggling them vertically or horizontally to express some exaggeration in movement.

Shearing: The shearing transformations to the vehicle images applied to the images to produce the perspective distortions.

This type of data augmentation methods takes place only during the training process, and hence it leads to a better model robustness in terms of variations in the looks and manifests as the improved detection accuracy and better generalization performance.

Color detection algorithms are the core for recognizing and classifying the vehicles according to their color characteristics. Vehicle colors most commonly serve as the main characteristic characteristics that help vehicles in diverse appearances especially from one model to another. The multi-colour detection approach allows the model to see and categorize vehicles in terms of the color nuances, leading to a higher degree of accuracy and precision in detecting driving vehicles.

Common color detection techniques used in vehicle model detection include:

Color Space Conversion: The translation of the RGB images of vehicles into various color spaces, for instance, HSV (Hue, Saturation, Value), LAB (Luminance, Green-Red, Blue-Yellow), or YCbCr (Luma, Chroma-blue, Chroma-red), to facilitate color segmentation and analysis.

Thresholding: Segment the vehicle regions by means of thresholding, region-based processing, and color, and select the needed colors for the purpose of the model.

Histogram Analysis: Vehicle images color histograms analysis to find its primary colors and characteristics with the aim of detecting picture based classification and recognition.

The model acquires color detection algorithms and integrates them into the vehicle model detection pipeline where the model can then identify colors of vehicles accurately to provide traffic officers, law enforcement and urban planners among others valuable identification information.

In short, functional transformations, as well as color detection approaches greatly improve the quality of car category acting systems Through the use of these techniques, deep learning models can get a better accuracy, robustness, and

versatility for the recognition and classification of vehicles in different real-world situations. The aim of this research is to examine the viability and efficiency of augmented reality and color recognition technologies for vehicle model identification, which will enhance the competence of the computer vision systems and improve the intelligent transportation.

1.2 PROBLEM STATEMENT

The modern-day vehicle model detection system is very crucial for traffic handling, surveillance system, and auto-security. This project will use state-of-the-art deep learning techniques such as the mobile net v2 architecture to develop a novel real-time model for detecting and classifying various car makes and models using images or videos.

Key Challenges:

Accuracy Across Diverse Conditions: Establish accuracy of vehicle model identifications for diverse environmental environments like many types of lightening, weather, as well as roads and surfaces.

Handling Scale and Perspective Variations: Handle spatial and scale challenges, among others, such as differences/variance between vehicle sizes in images and frames of video.

Optimization for Resource Constraints: Increase the model's efficiency to be run on machines with minimal computational capability but still accurate, and in real time.

High Precision Identification: Classify accurately various automobile brands and models in difficult conditions using high precision.

Real-time Performance: Create a quick model for processing with live identification of vehicles on live video streams.

Generalizability and Adaptability: Consider the robustness of the model in order to generalize for various geographies, vehicle types, and environments.

Enhanced Traffic Management: Create a tool to assist traffic monitoring systems with respect to congestion management and improving traffic movement.

Law Enforcement Support: This should help the police forces identify vehicles for reasons of security, surveillance, and forensics.

Intelligent Transportation Systems: Provide a solution for automated vehicles by helping in vehicle identification hence, assist in intelligent transportation system and other smart city initiatives.

Although deep learning algorithms for vehicles detection have been developed to a good extent, limited dataset labels and size lead to the lack of robustness and generalization capability of models.

The models that are currently in use are mostly trained on small datasets and therefore, they suffer from the overfitting and the bad performance when they are applied to real-world scenarios that include the variations in the lighting conditions, the viewpoints and the occlusions.

The purpose of statement refers to that analysis is performed to analyze if the rotation, scaling, flipping, and translation techniques of data augmentation contribute to increasing of the model detection models` performances or not in the situation of limited training data.

Identifying cars by their color attributes goes hand in hand with such uses as traffic monitoring, law enforcement and urban planning being one of them.

Universally existing vehicle model detection models, unfortunately, usually infer only visual information devoid of color feature, which constrains their capability to discern between identical-looking vehicles of distinct colors.

The problem statement is about the necessity of the creation and evaluation of color detection algorithms that can properly recognize and classify cars based on their color attributes which will improve the granularity and accuracy of the car model detection systems.

In spite of the fact that there are a lot of large-scale datasets that are rendered by the vehicle model detection, models that learn from these types of datasets are often not able to generalize to challenging scenarios of the real world with a lot of variations in the environmental conditions.

The augmentation methods are the possible solution. Use them to increase the diversity and richness of training data, which will lead models to learn the robust and invariant features that generalize well to unseen data.

The research problem evaluates how different enhancement techniques affect the precision and robustness of deep learning-based models for vehicle detection under different scenarios including various lighting and weather conditions and dense traffics.

Real-time automobile model recognition systems should have high inference speed and low computational load to be effective in changing conditions.

At the same time, classic methods might be prone to hike computational complexity during training thus prolonging training time and making more memory consumption.

The issue statement looks into the training of incredibly lightweight and effective augmentation strategies that could be used in vehicle shape detection, leading to faster training and less memory and processing power consumption.

The car color is a fundamental semantic feature that gives us important information for detecting and identifying the car model.

Nevertheless, the auto-model detection models existing, generally, put color as a secondary feature without considering its semantic meaning to the whole detection process.

The specific question to be addressed is using semantic color in the detection of vehicles' models through deploying a color detection technique to improve the model execution process, correctness, clarity, as well as vehicle features interpretation.

1.3 OBJECTIVES

The primary objectives of this project include:

1. Vehicle Model Detection: A deep learning model to determine the specific make or model of a car within an input picture. This involves training a diverse data set consisting of images of different vehicles' models.

2. Vehicle Color Detection: Building a tool capable of estimating the color, among other things of a vehicle from images. This involves teaching the AI model to identify differentiate between different vehicle colors.

3. Integration and Real-Time Detection: Enabling the development of an operational real-time system live video stream or image processing, vehicle detection, and real-time.
the model's prediction as well as the color prediction.

4. Accuracy and Performance: To obtain an accurate model with a strong reliability.
through suitable deep learning architecture, data preparation techniques.
evaluation metrics.

5. User-Friendly Interface: Creating an easy to navigate interface where users can upload documents and files is crucial.
vehicle model & color detection based on images or input video streams.

6. Application Scenarios: Using the system in different way such as traffic.
such processes include analysis, parking management, security surveillance and others.

7. Environmental Adaptability: Reinforce the model's robustness in identifying cars under the diverse environment circumstances such as diverse lightings, changeable weather, and obstructions.

8. Multi-Object Detection: Enhance the model's ability to identify multiple types/colors of automobiles in one photo or video snap.

9. Fine-Grained Classification: Provide detailed classification of unique alterations by model, variant, or series designation that will enhance accurate recognition.

10. Dataset Expansion and Diversity: Enhance the dataset progressively with new vehicle types, coatings, and background situations to boost generalizability and accuracy of the model.

11. Transfer Learning and Model Compression: To reduce the memory footprint, it should be possible to use other techniques such as transfer learning and model compression while maintaining high quality.

12. Localization and Object Tracking: Add an object location and follow-up capabilities to detecting vehicles that allows them to track their movements on subsequent frames for better traffic flow assessment and control.

13. Privacy Considerations: Privacy-conscious design features must be incorporated in the system while following data protection regulations regarding anonymization and/or managing sensitive information.

14. Model Explainability and Interpretability: Outline strategies for explaining the model's reasoning so that patients can trust prediction results needed for use in life saving purposes.

15. Robustness Testing and Error Analysis: Perform thorough validation of the model's behavior under extreme scenarios such as failure modes analysis and error correction algorithms.

16. Collaborative System Improvement: Incorporate user feedback in order to make incremental adjustments to improve the system's accuracy, reliability, and user friendliness.

17. Evaluate algorithms which increase the capacity towards differentiation and adaptation of detection models for vehicle recognition. Analyze the effect of rotation, scaling, flipping, and translation on the recognition models.

18. Find out how the techniques of augmentation can replace the lack of diversity and the size of the annotated datasets and hence make the models achieve better performance even with small training datasets.

19. Sketch augmentation parameters such as rotation angles, scaling factors, and translation distances that cater model efficiency to the point that it is neither overfitting nor underfitting.

20. Try inventing novel augmentation methods, e. g. elastic deformation, Gaussian noise injection and cutout regularization, to produce even more different and perfect datasets so that neural networks will acquire new skills and outperform on the tasks.

21. Assess Generalization Across Environmental Conditions:Assess Generalization Across Environmental Conditions

22. Examine the generalization capability of the augmented models across the various real-world scenarios, such as the variations in the lighting, weather, and occlusion, to make sure that they can perform well in the practical applications.

23. Evaluate the model performance with augmentations as well as baseline models and find the viability of such approaches. Write down the key takeaways from the study.

24. Analyze the feasibility of applying the learned methods for vehicle model detection with other tasks / objects / scenarios including pedestrian detection and traffic signs recognition. This will demonstrate the broader usability and usefulness of the learned methods.

25. Develop and implement color recognition algorithms which can be effectively used for identifying and sorting car colors based on their color phases, employing color space conversion, thresholding methods and histogram analysis.

26. Introduce algorithms that detect color into the vehicle model detection models, which in turn will use color information as an additional feature for the improvement of classification and recognition accuracy.

27. Examine how using color into vehicle model detection models yields accurate detection, precision, recall and F1-score metrics when compared to simpler models that use only visual features.

28. Check the semantic relevance of vehicle color attributes when ones are in the process of recognizing the vehicle models in which they perform their functional role. In this case, what role is the color information playing

1.4 SIGNIFICANCE AND MOTIVATION

The project has the potential to offer several benefits:

1. Automation and Efficiency: It is possible for the system to automate the identification of vehicle model types and colors which will eliminate human intervention.

2. Data Collection and Analysis: This helps to understand patterns of consumers who share and respond to information of vehicle patterns, preferences, and trends.

3. Enhanced Security and Surveillance: Security is guaranteed as it enables people to log in their personal accounts and ensures fast detection of the vehicle in the prohibited zone or a suspicious car.

4. Traffic Management: Better traffic management is a result of vehicle detection and analysis for managing and optimizing road infrastructure.

5. User Convenience: The system allows users to access data easily enabling one search for information fast and with ease without a manual inspection.

6. Accident Analysis and Forensics: Provide quick data to support post accident analysis, insurance claims, and help in forensic investigations.

7. Parking Solutions: To detect and follow vehicles inside parking lots, allocate space effectively, and offer immediate occupancy reports for parking management systems.

8. Customized Marketing and Services: Using this information, targeted marketing strategies and personalized services can be enabled through the analysis of the car model data and customer preference records taken by the system.

9. Fleet Management and Logistics: Auto-identify vehicles, monitor them, route them well, and help ensure that they follow transportation rules in order to support effective fleet management.

10. Urban Planning and Infrastructure Development: Useful information to the urban authorities regarding road enlargement, traffic conditions improvement and other related infrastructures based on patterns of car movements.

11. Public Safety and Emergency Response: Supporting emergency response vehicles by pinpointing cars in distress or emergency zones and fast tracking help when needed.

12. Vehicle Theft Prevention: Help law enforcement officials in combating vehicle theft by making it easier for the same to apprehend stolen vehicles.

13. Environmental Impact Assessment: Assessment of emissions and environment impact from vehicles type and quantity.

14. Enhanced Customer Experience: Enhancing customer satisfaction through quick and accurate delivery of vehicle information to improve automotive sales and service .

15. Legal and Compliance Assistance: Help police officers implement traffic rules, track faulty cars and promote safe driving.

The field of computer vision and deep learning is rapidly evolving, offering an opportunity for redefining automotive safety, surveillance, and efficient operations. The aim is to leverage the strength of a new mobile-optimized deep learning architecture—MobileNetV2, for a novel vehicle model detection system. This provides a reliable, fast, and real time solution that correctly identifies numerous different makes of vehicles. It is through this innovation that the project hopes to empower critical applications ranging from traffic monitoring, law enforcement, to intelligent transport systems. The objective is to enhance security, expedite vehicle monitoring, and boost intelligent transport systems by providing a dependable and real-time tracking system.

16. Enhanced Model Robustness: The augmentation techniques are having a much important job in increasing the strength of the vehicle model detection networks. Through augmenting the model with the data conclusions as diverse as viewpoint alterings, lighting condition modifications, and occlusion changes, the model becomes more competent in making generalizations that do apply to the unseen scenarios.

17. Mitigation of Data Scarcity: The annotated datasets for vehicle model detection are usually small and diverse in size. As opposed to humans where we could only manually label a tiny portion of data, technology offers an "augmentation" solution to this challenge. This solution allows the model to learn a richer set of features, thus improving its performance even when the limited data is labeled manually.

18. Improved Generalization: This allows models trained on these data to generalize better to scenes of real-life scenarios, including a variety of environmental conditions and instances where the encountered vehicle data may have never been seen before. Augmentation enables the model to learn the invariant features that are robust to such variations, and thus, the model can rely on these features when it is deployed in real-world scenarios.

19. Optimized Model Training: Techniques of data augmentation prevent overfitting and underfitting by showing the model after a composite of the broader shade of data variations is being trained. The model is continuously refined in such a way that it not only absorbs the main features of vehicle models but also inherent randomness of the dataset without simply memorizing specific representations from the training data.

20. Efficient Resource Utilization: Augmentation serves as a technique of making the best use of the existing annotated data as it allows for the more efficient utilization of computational resources. Rather than enlarging manually annotated data sets, the technique of augmentation uses synthetic data obtained from available sources, in toto saving on manpower resources.

21. Semantic Understanding of Vehicles: Color is a vital semantic element of vehicles that gives valuable information about different car models and types. One way to do this is to

22. embed the color detection technique into the vehicle model detection models, so this will enable the model to comprehend semantically the car attributes and boost the detection accuracy.

23. Fine-grained Classification: Hue detection works as a precision instrument measuring vehicle color grades for highly specific sampling providing detailed and reliable classifications of vehicles. This refined classification can be very useful in areas like traffic monitoring, law enforcement, and urban planning.

24. Improved Discriminative Power: This contrastively enhances the unique discriminating power of visual features in vehicle model detection, which is something extra. The utilization of color attributes will help the model to tell the difference between the same-looking vehicles but with different colors, thus obtaining more accurate and faithful output.

25. Environmental Adaptability: The algorithms for color detection are immune to variations in environmental conditions like changes in lighting, weather conditions, and occlusions. The inclusion of the color detection module in the model of vehicle detection system is an attempt to make the model design adaptive in various real life situations by improving its performance in conditions which alter its ability to detect.

26. Enhanced Interpretability: The use of features that are based on color allows for understandable outputs, the decision-making process and the predictions that could be made could be easily explained. Through the study of the color attributes, we can get the idea of the model's working procedure and thus we can enhance the interpretability and explainability of the vehicle model detection systems.

1.5 ORGANIZATION OF PROJECT REPORT

Chapter1- Discussed about the goals of project, the problems it targets, the objectives it has and the significance and motivation behind the project.

Chapter2- Discussed about the limitations of various research papers and the literature survey.

Chapter3- Discussed about the requirements for the project like hardware, software, dataset, performance, functional, non-functional, The project design and it`s architecture, components data preparation, implementation, tools and technologies used and the challenges encountered.

Chapter4- Discussed the testing phase.

Chapter5- Discussed about the results we got after the successful performance of the model.

Chapter6- Discussed about the conclusions derived from the model`s result and the future scope of the project.

CHAPTER 2: LITERATURE SURVEY

2.1 OVERVIEW OF RELEVANT LITERATURE

S. No.	Paper Title [Cite]	Journal/ Conference (Year)	Tools/ Techniques/ Dataset	Results
1.	A Fast and Accurate Real-Time Vehicle Detection Method Using Deep Learning for Unconstrained Environments	2023	YOLO-v5 architecture for vehicle detection and classification . Transfer learning, PKU dataset	The research presents YOLO-v5—a quick, precise, real-time vehicle detection technique—for unrestricted contexts utilizing publicly accessible datasets.
2.	Vehicle Model Prediction Using Machine Learning and Deep Learning.	2023	Linear regression, Support Vector Machines (SVM), K-Nearest Neighbors (KNN) and Convolutional Neural Networks (CNN), car images dataset	98.5% average accuracy,with 29% and 20% accuracy from linear regression and K-Nearest Neighbors (KNN), rendering them unsuitable for intricate vehicle make and model recognition.

S. No.	Paper Title [Cite]	Journal/ Conference (Year)	Tools/ Techniques/ Dataset	Results
3.	Vehicle Detection and Recognition	2020	Principal Component Analysis , Hough transformation and segmentation techniques , Data preprocessing techniques	Using the model developed by Tang Yang and his team, the average accuracy of vehicle detection is 97.
4.	Vehicle Classification and Detection using Deep Learning	2019	ConvolutionalNeural Network (CNN) , Fast R-CNN, Camera, BIT-Vehicle dataset	The suggested network predicts bounding boxes for automobiles, bikes, and humans with greater accuracy than Faster R-CNN, particularly for motorbikes.

S. No.	Paper Title [Cite]	Journal/ Conference (Year)	Tools/ Techniques/ Dataset	Results
5.	Image-based approach for vehicle model re-identification	2019	modified version of the AlexNet neural network, comparison techniques, siamese networks	This study presents a low-complexity similarity criterion and a modified AlexNet for feature extraction in an image-based car model re-identification system with competitive accuracy.
6.	Vehicle type detection based on deep learning in traffic scene	2018	RCNN framework, RPN (Region Proposal Network), Convolutional Neural Network (CNN) algorithm, MIT and Caltech car datasets are combined.	With experimental backing, the research confirms that Faster RCNN with enhanced RPN networks is effective for traffic scene vehicle type detection.

S. No.	Paper Title [Cite]	Journal/ Conference (Year)	Tools/ Techniques/ Dataset	Results
7.	Vehicle detection and tracking using ML techniques	2023	Support Vector Machine (SVM) and Decision Tree algorithms and python programming	For the purpose of tracking and detecting vehicles, the study contrasted SVM and Decision Tree. SVM performed more accurately than Decision Tree.
8.	Vehicle Detection and Type Classification Based on CNN-SVM	2021	YOLO deep learning series algorithm , CNN was combined with SVM	Particularly for "van" and "others," enhanced AlexNet with SPP outperformed the original in recall. Using SVM improved recall, showcasing its improved accuracy fine-tuning and generalization.

S. No.	Paper Title [Cite]	Journal/ Conference (Year)	Tools/ Techniques/ Dataset	Results
9.	Vehicle model identification method based on convolutional neural network	2019	Data enhancement, Adam optimization, Transfer learning	The technique makes use of transfer learning, an AM Softmax loss function, and an optimized CNN (data augmentation, Adam) to improve accuracy in a variety of backgrounds.
10.	An Effective Approach of Vehicle Detection Using Deep Learning	2022	YOLOv3 algorithm and the SSD algorithm, BDD100K dataset	In evaluating tracking, segmentation, and autonomous driving models, the paper compares the vehicle recognition algorithms of SSD and YOLOv3.

S. No.	Paper Title [Cite]	Journal/ Conference (Year)	Tools/ Techniques/ Dataset	Results
11.	DATA AUGMENTATION ANALYSIS OF VEHICLE DETECTION IN AERIAL IMAGE	2023	modern techniques such as Fast R-CNN, Faster R-CNN, Mask R-CNN and YOLO3, SSD, and RetinaNet and AERIAL and XDUAVdatasets.	Explored the influence of distinctive data augmentation methods on aerial image object detection by mainly focusing on the UINAERIAU and DRUAV datasets
12.	Generative Data Augmentation for Vehicle Detection in Aerial Images	2021	Improving vehicle detection in aerial images with a small amount of training data via generative augmentation and VEDAI Dataset.	Incubation of network parameters performed by 1000 times augmentation 500 images for best coverage threshold and IoU for highest quality of images.

S. No.	Paper Title [Cite]	Journal/ Conference (Year)	Tools/ Techniques/ Dataset	Results
13.	Real-time Vehicle Detection implementing Deep Convolutional Neural Network features Data Augmentation Technique	2022	Faster Region-based Convolutional Neural Network (R-CNN),a pre-trained deep model VGG-16, speed and data augmentation, custom vehicle dataset having a combination of 3192 images	The original hard work based method came up with a Sensitivity value of 93. 5% and an accuracy value of 87. 6% within 0. 42s. The profit from the development of personalized data emerged a sense of growth by 4% and cutting off response time by 3. 23s.

2.2 KEY GAPS IN LITERATURE

1. The fact that there are restrictions on the number of good quality labeled training samples that affect the accuracy of single vehicle detection methods.

2. Data set and vehicle model recognition is not included in the paper and no information about the problems associated with implementation and limitations for the machine and deep learning based detection system.

3. Camera placement and coverage are key to hybrid model security because camera vulnerabilities may cause system insecurity.

4. This paper disregards image noise, only studies pictures of cars, cycles, and busses instead of many other things, and there is no discussion about the network's computational complexity or expenditure needs.

5. There is no discussion on the effects of lighting, weather, and occlusion on accuracy; there is also nothing on computational requirements and scalability that can make this work useful in practice.

6. N.A.

7. The detailed result and analysis of algorithm performance are not adequately discussed in the paper. Also, it should contain more objects in both image classes from the dataset to give better results.

8. There is no comparative analysis of other methods, and information about the dataset (e.g., in terms of the number of participants, representativeness, and characteristics) is not enough.

9. This paper is not well-founded; it has no thorough evaluations, comparisons, data enhancement for validation, and doesn't address real-world challenges or generalizable to other types of vehicles or models.

10. There are two shortcomings associated with YOLOv3 and SSD, namely lighting and weather impacts in real time. Without hardware details, the paper's models do slightly poorer over the board, particularly for buses and motors.

11. Object detection in aerial pictures is confronted with a number of challenges which include the issue of size variation, different distances, low image object sizes. The aerial images have small objects that are limited by low pixels, which makes the feature extraction process very hard. The paper is dedicated to vehicle object detection that may miss some objects other than vehicles such as humans, horses, and trees, which in turn impede the completeness of the findings. The research identifies the difficulty in designing an object detection model that is trained with an aerial image of a limited dataset for effectively implementing the suggested data augmentation techniques

12.. The routes taken in this paper are centered on the single class of instance (cars) on cross-view photos, which can be found a difficulty of expanding the multiclass situation of detection problem. It does not deal extensively with other varieties of objects or descriptions of events that this method might use.

13. The scheme's effectiveness may be hampered by algorithm defects and loss of classification accuracy. . The research is conducted through a customized dataset with 3200 images, which can potentially have an effect on the generalizability of the results to other datasets such as PASCAL-VOC, KITTI, and COCO. Results may differ when a dataset is used as a real-life starting point. The system's accuracy score can be increased by trying out other methods of calculations. This is not a framed study and hence, there is no use of open datasets to show if the system is effective or not, and therefore, the findings cannot be generalized.

CHAPTER 3:SYSTEM DEVELOPMENT

3.1 REQUIREMENT AND ANALYSIS

The goal of this research is to create a vehicle model detection system using the MobileNetV2 architecture, a convolutional neural network (CNN) that is effective and lightweight and can be used in real-time applications. The goal is to develop a reliable and computationally effective method for recognising and categorizing different vehicle models using picture and video data. With a focus on utilizing the MobileNetV2 architecture for efficient feature extraction and classification, the project covers important phases like data collection, preprocessing, model training, and evaluation.

Some requirements are:

Diverse Augmentation Techniques: Deploying a blend of enhancement methods including rotation, scaling, flipping, translation, and shearing is the objective in order to provide the augmented dataset with a broad sampling of car images shot from multiple viewpoints.

Parameter Tuning: Refine the augmentation parameters such as rotation angles, scaling factors and translation distances to prevent overfitting and underfitting simultaneously and for the sake of reaching the desired level of model performance.

Efficient Implementation: Setting up the augmentation pipelines that are compatible with the model training frameworks is the way to reduce the computational overhead and the training time.

Compatibility with Annotated Data: Make sure augmented images do not damage bounding boxes or segmentation masks drawn in so that they remain true during training and the use of ground truth annotation in training remains accurate.

Robustness to Environmental Conditions: When developing these systems, take into account integral modifications of the conditions of weather and lighting and obstruction techniques in order to promote the models' accuracy and ability to generalize variability.

Effective Color Detection Algorithms: Create true and efficient color detection algorithms with high accuracy that can classify the vehicles into categories depending on their color attributes and lower the error rate, these attribute detection algorithms should be entirely reliable.

Robustness to Variations: Develop algorithms that can be capable of recognizing colors even when the conditions of the environment such as the lighting change, the weather condition is different or something is blocking the view.

Semantic Understanding of Color Attributes: In this analysis, compare the semantic significance of color attributes in the automobile model detection domain, and find out how the color information helps in making interpretable and explainable models of vehicle elements, and also assists in understanding the semantic meaning of vehicle features.

Some analysis are:

Impact on Model Performance: Conduct an assessment of augmentation techniques effectiveness on performance metrics including precision, recall, accuracy and F1-score, with determining these metrics via models trained with augmented data.

Generalization Across Environmental Conditions: The practicality of the augmented models' generalization ability across varied real-world scenarios such as the changes in lighting, weather, and occlusion is to be ascertained to ensure their robust performance in the usage settings.

Transfer Learning Potential: Conduct a research by applying learnings from datasets obtained by augmentation to some related detection activities like pedestrian detection or traffic sign recognition in order to comprehend whether the transferability of features is possible to broader scenarios which use the same learning methods.

Scalability and Efficiency: The scalability and efficiency of augmentation techniques in dealing with large-scale datasets and computational resources, the training time, memory consumption, and computational overhead shall be analyzed.

Impact on Annotation Efforts: Explore the role of augmentation in annotation, checking if the presence of an augmented dataset needs extra manual verification or particular tuning of ground truths in order to guarantee the accuracy and uniformity of the annotation.

Comparison with State-of-the-Art Approaches: Compare the dynamics of the augmented classifiers and the current best practices in vehicle detection, pointing out the accuracy, effectiveness, and precision trade-offs.

Impact on Model Performance: Assess the role of the color detection algorithm on model performance measures to examine the effect of the color detection algorithms on accuracy, precision, recall, and F1-score while analyzing the results with the visual based models.

Fine-grained Classification: Examine the efficiency of color-based classification methods in separating various car models and types by the features of their colors, finding out the level of precision and accuracy of the classification results.

Robustness to Environmental Conditions: Perform tests to evaluate the dependability of color-based feature detecting algorithms against lighting variations, weather conditions, and occlusions. Determine, if the detecting algorithms are applicable in real-world scenarios providing stable and reliable cues to envision automotive models.

Integration with Visual Features: Explore how color information joins visual attributes in detection of the model of the vehicles and see whether the integration of color signal leads to more precise, robust and generally applicable detection.

Interpretability and Explainability: The color-based features of the color-based features are studied in the vehicle model detection models and the explanation of the decision-making process is provided and the patterns and characteristics of the vehicle models are given.

3.1.1 HARDWARE REQUIREMENT

- GPU-powered hardware to train models more effectively.
- cameras or other image-input devices to take pictures of cars.
- enough storage space to hold trained models and datasets.

3.1.2 SOFTWARE REQUIREMENT

- Model development using a deep learning framework (such as PyTorch or TensorFlow).
- OpenCV image processing libraries are used for preparing data.
- Python programming environment for coding and scripting.
- Development environment compatible with MobileNetV2 architecture.

3.1.3 DATASET REQUIREMENT

- A comprehensive dataset with pictures featuring various car models.
- Vehicle locations and associated labels shown in annotated dataset with bounding boxes.

3.1.4 PERFORMANCE REQUIREMENT

- Processing power in real time for analysis of live video feeds.
- high precision and recall metrics, together with high accuracy in detecting vehicle models, all in line with project goals.
- Scalability to manage diverse environmental conditions and large datasets.

3.1.5 FUNCTIONAL REQUIREMENT

- **Data preprocessing:**
  - Resizing and standardizing pictures to meet the input specifications of MobileNetV2
  - Methods for enhancing dataset variety through augmentation

- **Model Development**:
  - Implementation of MobileNetV2 architecture for feature extraction
  - Transfer learning with pre-trained weights to leverage general image features.
  - Fine-tuning on the specific vehicle model dataset.

- **Training and Evaluation:**
  - Evaluation using metrics such as precision, recall, and F1 score
  - Efficient training with GPU acceleration

3.1.6 Non-functional Requirements:

- **Performance:**
  - The model should achieve a high level of accuracy in vehicle model detection.
  - Inference speed should meet real-time processing requirements.

- **Scalability:**
  - The system should handle variations in dataset size and environmental conditions.
  - Efficient resource utilization is necessary for scalability to larger datasets.
- **Usability:**
  - User-friendly interfaces for dataset annotation and model training.
  - Documentation for easy integration into different environments.

- **Reliability:**
  - The system should exhibit robust performance under varying lighting and weather conditions.
  - Adequate error handling and logging mechanisms

3.2 PROJECT DESIGN AND ARCHITECTURE

A well-structured and solid design and architecture of the project is built to detect models in vehicles using mobile version two.This is done through data flow where unprocessed images or video recordings from cameras constitute as inputs. The preprocessing module resizes, normalizes and augments the data so that the model may be applied in different circumstances. The most important part of this is that the MobileNetV2 model implements the transfer learning using pre-trained weights which are used in the features extraction as well as in the special tuning of the car detection model. Efficiency when using general visual attribute specifics like colour, size, and shape of car. The latter rounds it up with the output presentation part involving bounding boxes around the identified cars followed by label each model respectively.

3.2.1. Overview:

Objective: Come up with an effective mobile vehicle model detection system adopting MobileNetV2 architecture aimed at speedy and accurate operations.

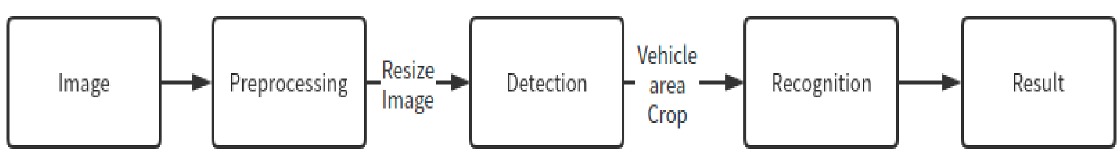


Fig 1: Design of Model

3.2.2. System Architecture:

Data Flow

Input Module: It also collects the data from various cameras including the video streams and pictures captured. The system receives feeds directly from these sources.

Preprocessing Module: The preprocessing module acts upon receiving data from the input module. It essentially prepares the incoming data for training and inference purposes. This involves several steps:
Rescaling: Standardization of all images in different formats to a specific size or resolution. It provides for consistency in processing.
Normalization: Scaling the pixel values of the images to a normalized scale, e.g., 0-1 for training model convergence and stability.
Augmentation: Using different variants and tweaks for the data in order to achieve a more generalized model across various entities. The augmentation techniques can be flipping, rotating, zooming, changing brightness and contrast among them.

Model Inference Module: It utilizes MobileNevv2, which is the lightweight variation of V2-ResNeXt neural networks tailored for smartphones and intelligent sensors. Here it is used as a detector for vehicle model in the pre-processed. The architecture of mobileNev2 is well known for balancing accuracy versus computational time so that it can work smoothly in applications that require real-time such as this one.

Output Module: After the Model Inference Module accomplishes its duty, the out module converts the detected car models into readable form for a human. Most often this implies drawing outlines around the identified cars and stating the car's make or model. Presenting in order to make it big.
clearly, present this information in a visual form, perhaps using enlarged or highlighted bounding boxes and captions for easy recognition.

The whole data flow pipeline is vital for the continuous search and display of vehicle models based on the camera inputs in different uses such as security surveillance, urban traffic monitoring, or object characterization programs.

Components

Data Preprocessing Component:

Resize and Normalization:

Resize: This helps to ensure that all images are made uniform in terms of size or resolution. Having consistent size of an image allows a machine to quickly process the data.

Normalization: Scalar normalization (normalizing the pixel values of images down to a standard scale, e.g. between 0 and 1 or –1 and 1). Normalisation promotes robustness of training processes by avoiding high levels of variation among pixels, which could overshadow the whole learning process.

Data Augmentation Techniques:

Variety Enhancement: The process of data augmentation entails adding or making changes to the dataset. The procedures include inversion, turning over, rotation, zooming, trimming, as well as tweaking on the brightness/contrast yield diverse occurrences of the same picture enhancing the data set.

Robustness Improvement: Augmentation ensures that the model can deal with a wide range of situations such as orientation, illumination, viewpoint, and scenarios in a real deployment environment.

Importance of Data Preprocessing:

Generalization: This model acquires sufficient diversity through performing rescaling, normalization and data augmentation which gives it robust features. The act enables the models to function appropriately in diverse natural settings they experience in the world.

Adaptability: Preprocessing involves showing the model different scenarios it is not accustomed to that it had never seen before during training. Hence, it adjusts when used in different aspects including varied lighting conditions, angles, and perspectives hence identifying features easily.

Reducing Overfitting and Improving Stability:

- Overfitting Mitigation: Pre-processors are helpful in making sure that the model does not fit itself on certain traits observed in the trained data. The technique of augmentation includes some modifications, which make it difficult for the model to remember particular cases but learn distinctive and comprehensive characteristics instead.
- Enhanced Stability: Normalization helps to normalize the training process by scaling the received input. This helps optimization becomes more uniform when training a model.
- Facilitating Efficient Model Training:
- Consistency in Input: Preprocessing standardizes the input data such that the model is trained on well-aligned data, which makes it possible for the model to converge faster.

MobileNetV2 Model Component:

Architecture:

Uses the MobileNetV2 architecture for feature extraction. Specially fine-tuned for automobile image classification and object detection.

Training and Inference:

They were trained on annotated dataset for model learning.

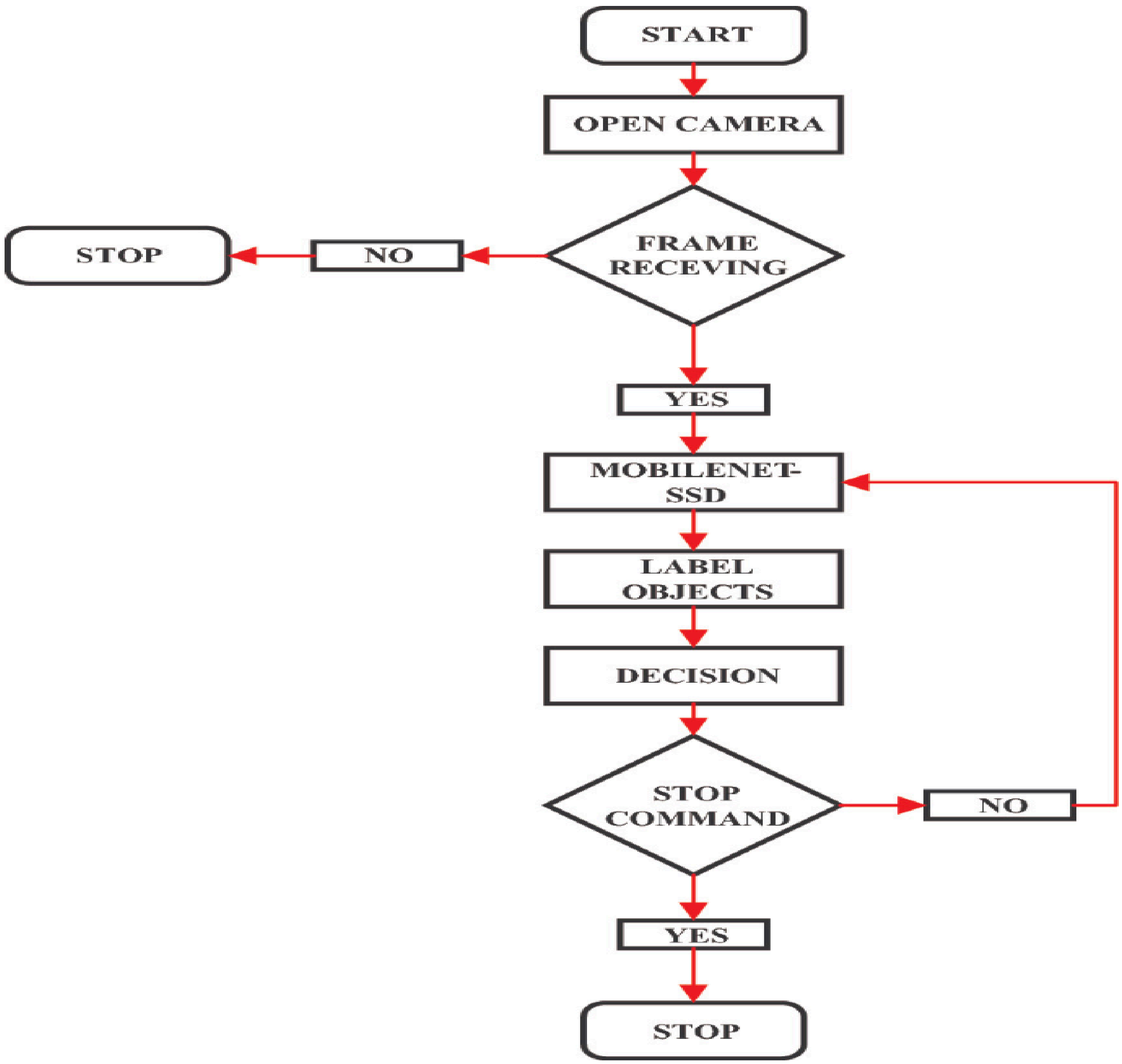


Fig 2: Architecture of MobileNetV2

3.2.3. Development Stages:

- Data Collection and Annotation: Bounding box annotated diverse set of vehicle images for several models.
- Data Preprocessing: Apply the pre-processing steps of image resizing, normalization, and augmentation on the data set for training.
- Model Training: Utilize MobileNetV2 architecture for training. Train the model using the annotated vehicle dataset.
- Model Evaluation: Evaluate the model's performance with measures such as precision, recall, and F1-score. Based on the evaluation, refine the model.

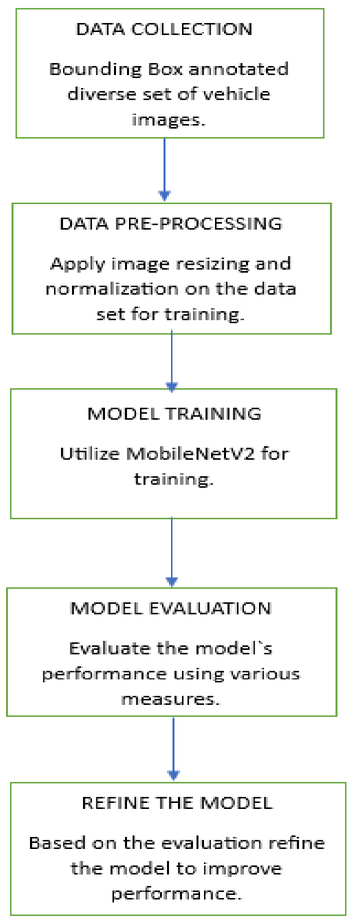


Fig 3: Development Stages

3.3 DATA PREPARATION

Data set preparation for vehicle object detection using MobileNetV2 is one of the important stages in development process, which lays a ground for performance of a model concerning accuracy and generalization. It is imperative that the dataset should cover an array of models of cars captured under different conditions of environment. The dataset of images should have bounding boxes that denote the location of cars alongside labels specifying the car model for each image. It also ensures that equal attention is paid to ensure balance in vehicle types and variations per model. Images are resized for compliance with MobileNetV2 input criteria and pixel values normalized to aid model convergence while data augmentation techniques such as rotation and image flipping are employed to expand variations within the dataset. Therefore, data set preparation has become one of the essential factors leading to success in this project since dataset quality and diversity will determine how well a model will be able to identify and differentiate car types accurately.

```
#data preprocessing
image_datagen = ImageDataGenerator(
    preprocessing_function=preprocess_function,
    rotation_range=5,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.1,
    horizontal_flip=True,
    validation_split=0.1,
    fill_mode='nearest'
)
train_generator= image_datagen.flow_from_directory(
    train_dr,
    target_size=(image_size,image_size),
    batch_size=batch_size,
    class_mode='categorical',
    subset='training'
)
train_images_count=len(train_generator.filenames)

validation_generator = image_datagen.flow_from_directory(
    train_dr,
    target_size=(image_size, image_size),
    batch_size=batch_size,
    class_mode='categorical',
    subset='validation'
)
validation_images_count = len(validation_generator.filenames)

with open('classes_name.pickle','wb') as handle:
  pickle.dump(validation_generator.class_indices,handle,protocol=pickle.HIGHEST_PROTOCOL)
```

```
Found 2806 images belonging to 48 classes.
Found 290 images belonging to 48 classes.
```

Fig 4: Data Preprocessing

3.4 IMPLEMENTATION

The model is made to load an image that is then resized in order to fit the input dimensions. The loaded image is transformed into a number of array so as to be used for input. predictions made on this loaded imaging are made by use of already trained model. The image is transformed to fit the model's expected input structure, and the model.predict() function is applied to get class probability scores.Based on the prediction results, the top-rated classes as well as their respective probabilities are extracted from the prediction scores. The findings are subsequently presented in an organized manner by pairing each class label with its corresponding probability of occurrence.

Here, we present our step for making predictions from one input image using a trained deep learning model. The code demonstrates how one can use the developed vehicle model and color vision system by loading the model, preprocessing the image, generating predictions and interpreting the obtained results. This is a vital section in the entire project that enables users to understand how to classify vehicle models and colors from images of different vehicles in the country.

```
from keras.applications.mobilenet_v2 import preprocess_input
train_dr= "path to training dataset"
train_images= list(paths.list_images("path to training dataset"))
num_classes=len(os.listdir(train_dr))
def get_mobilenetv2_full_tune_model_alpha_1_4_concatenated_regularised(_num_classes) \
        -> Tuple[Model, Model, int, Callable]:
    image_size = 224
    channels_count = 3
    initial_model: Model = MobileNetV2(weights='imagenet', alpha=1.4, include_top=False,
                                       input_shape=(image_size, image_size, channels_count))
    initial_model.trainable= True
    for i,layer in enumerate(initial_model.layers):
       layer.trainable=True

    initial_model_output=initial_model.output
    x=layers.GlobalAveragePooling2D()(initial_model_output)
    regularizer = regularizers.l2(0.01)
    x=layers.Dense(1024, activation='relu', kernel_regularizer=regularizer)(x)
    predictions = layers.Dense(_num_classes, activation='softmax')(x)

    model=Model(initial_model.input, predictions)
    return model, initial_model, image_size, preprocess_input
```

Fig 5: Model Implementation

```
def get_callbacks_list(_early_stopping_patience, _reduce_lr_on_plateau_factor,_reduce_lr_on_plateau_patience):
    return[
        keras.callbacks.EarlyStopping(
            monitor='val_acc',
            patience=_early_stopping_patience
        ),
        keras.callbacks.ModelCheckpoint(
            verbose=1,
            filepath='best_model.h5',
            monitor='val_loss',
            save_best_only=True
        ),
        keras.callbacks.ReduceLROnPlateau(
            verbose=1,
            monitor='val_loss',
            factor=_reduce_lr_on_plateau_factor,
            patience=_reduce_lr_on_plateau_patience
        )
    ]
```

Fig 6: Model Implementation

Tools and Technologies:

- Deep Learning Frameworks: Use TensorFlow or PyTorch for building and running the model.
- Image Processing Libraries: Data preprocessing and manipulation using open CV.
- Deployment Frameworks: The deployment of a web service can be performed with TensorFlow Serving, ONNX Runtime or flask.
- TensorFlow

  -Overview: TensorFlow is a free deep learning framework created by Google which helps in the construction and development of machine learning models The environment provides a full set of instruments and libraries for different objectives.

  - Key Features:
    - Flexibility: In addition, tensorflow offers high-level APIs for fast model prototype(such as Keras), low-level APIs for fine tuning and customization.
      -
    - Scalability: It can be deployed over a number of devices – from small portable devices like mobile to large scale disperse systems.
      -
    - Community and Documentation:Beginner-friendly – there is a growing community and vast documentation on TensorFlow.

- PyTorch

  - Overview: Another widely used open-source deep learning framework was designed by Facebook which is called PyTorch. This is referred to as PyTorch because of its dynamic computational graph, making it more preferred than other tools in research and experimentation.

    - Key Features:
      - Dynamic Computational Graph: PyTorch uses a dynamical computational graph that favors such tasks as dealing with varying or dynamic databases of information.

      - Ease of Debugging: PyTorch has a more Pythonic and imperative syntax making debugging as well as introspection easier when developing models.

-

      - Research Focus: It is an easy to use platform and popular across the research community.

- OpenCV (Open Source Computer Vision)

  - Overview: OpenCV is an open-source computer vision library with many tools and algorithms for image processing and applications in computer vision.

  Key Features:

  - ◆ Extensive Functionality: Some example operations in this regard that OpenCV facilitates include rescaling, normalization, augmentation etc.

- ◆ Cross-Platform: It is compatible with various platforms like windows, linux, macOs, android and Ios which makes it adaptable in several deployment scenarios.

- ◆ Community and Integration: Providing large community support for integration with a wide range of other famous libraries and frameworks.

- ● TensorFlow Serving

  - Overview: Specially designed serving system for deploying ML models especially those formulated with TensorFlow.

  Key Features:

  - ◆ Scalability: Built for serving TensorFlow models at scale – suitable for production settings.

  - ◆ REST API: It allows for an open exposure of models through a REST API, thereby making it easy to integrate with multiple applications or systems.

- ● ONNX Runtime

  - Overview: ONNX Runtime is a free and open framework and open inference engine for the Open Neural Network Exchange (ONNX) format, an open standard for AI models.

Key Features:

- Interoperability: Provides support for ONNX which is a platform that trains models on different frameworks such as PyTorch, TensorFlow, or scikit-learn.

- High Performance: Well optimized to run efficiently on different kind of hardware such as CPUs and GPUs.

- Flask

- Overview: Flask is a simple python based web framework, which works perfectly in developing web apps and APIs.

Key Features:

- Ease of Use: One can easily set up and use Flask making them fit for small scale deployments.

- Flexibility: Flexible design of API for RESTful service and machine learning model serving.

- Community and Extensions: It has an interesting community and a modular structure that makes it easy to integrate with different extensions for additional functionality.

1.) Keras

Overview- Keras is a high-level neural networks API written in Python and behaving as a layer top of TensorFlow , Theano, Microsoft Cognitive Toolkit (CNTK). It is a work that has been created with the aim of quick experimentation and easy prototyping of deep learning models

key features:

User-Friendly Interface:Keras is having an easy-to-use and intuitive API wherein a beginner can just conceptually design deep learning models and train them with no or little coding involved

Modular and Extensible Architecture:Keras employs a modular design philosophy, where individuals can easily create neural network models by combining reusable blocks called layers.

Compatibility and Backend Support:Keras provides a smooth and hassle-free coexistence of different backend engines, such as TensorFlow, Theano, and CNTK.

2.) Matplotlib

Overview-is a complete collection of libraries for making ready, animated and sharing of explorative visualizations in Python. It is also a complete solution that helps in making plots, charts, histograms, and other graphical tools to organize data.

Key features:Simple and Intuitive API:Matplotlib provides a simple and user-friendly API for the creation of a broad range of charts and visualizations with only a few lines of code.

Multi-platform Support:Matplotlib is a multi-platform package, it supports all the major operating systems- Windows, macOS and Linux.

Versatile Plotting Capabilities:Matplotlib offers a variety of plot types such as line plots, scatter plots, bar plots, histogram plots, pie charts, heatmaps and more.

3.) functools

Overview- the Python module holds higher-order functions and utilities working with functions and execution objects.

key features:

Decorators:functools contains the wraps() decorator that is used to make well-behaved decorators.

Higher-Order Functions:functools provides several higher-order functions, which are namely the partial(),reduce() and singledispatch() which allows functional programming.

Function Composition:The partial() function is a kind of function application that allows for the fixing of a subset of a function's arguments and the return of a new function with the remaining arguments.

4.) Pickle

Overview- The pickle module in Python is used for serializing and deserializing Python objects to save them to a file or send them over a network.

Key features:
Serialisation and Deserialization:pickle is the toolkit which offers its two functions - pickling Python objects into binary format and unpickling back to their Python object forms.

Platform-Independent:The Pickle files are platform-independent; they can be created on one system and read on another without compromising the compatibility.

Support for Arbitrary Python Objects:One of the unique features of Pickle, which really stands out among others, is its marshaling ability.

5.) Path

Overview- In the world of Python, the word "path" is synonymous to representing the path that the file or directory goes through in the system file. Among others, Python library is not directly provided with a named "path" but the pathlib module is more widely utilized to give classes that work with the file path in a way that is not platform-dependent and object-oriented.

Key features:

Object-Oriented Approach:pathlib is a class-based paradigm for representing file paths and working with the file system.

Platform Independence:In pathlib, projects are abstracted away from the platform-specific syntax of file paths

Concise and Readable Syntax:pathlib provides a compact and clear code for file path building, accessing and manipulating.

6.) Torch

Overview- created by Facebook's AI Research (FAIR) group and known as HomeBrew, which is for Python programming language. The transports that are necessary to build and train the deep learning framework stand out as very flexible and efficient. Here's an overview and key features of PyTorch:

Key features:

Dynamic Computational Graphs:The most crucial aspect of PyTorch is its capability to deal with dynamic computational graphs.

Numpy-like Interface:PyTorch offers a Numpy-like interface for tensor manipulation, which makes it simple to switch from Numpy to PyTorch.

GPU Acceleration:Pytorch offers an easy way to take advantage of GPUs without having to think about hands-on with them.

3.5 KEY CHALLENGES

There are various difficulties encountered during the entire period of developing a vehicle model detection system. To ensure the resilience and feasibility of the proposed solution, then it is vital to address these challenges.

Limited and Diverse Dataset:

Challenge: It is difficult to get a diversified collection with many samples of models that are not frequent such as the rare ones.

Addressing: Engaging in an active collection of data across cooperation with automakers, image catalogs, or crowdsourced activities. Techniques of augmentation enable increasing the variety of dataset, while the experience obtained from a larger set is used for transfer learning.

Computational Resources:

Challenge: Training deep learning models, particularly complex architectures such as MobileNetV2, is quite computationally expensive and might pose constraints to some developers.

Addressing: Provision of cloud based services with GPU support or utilization of pre-trained models will assist in reducing the workload. Also, there are model quantisation and optimize techniques that could be used to reduce the model's size while ensuring that performance is not affected much

Real-world Variability:

Challenge: In many different environments, where one might encounter light conditions like fog, rain, or occlusion, vehicles may look completely different than what they really are.

Addressing: Using advanced data augmentation methods during dataset preprocessing to mimic different scenarios. Utilization of sophisticated image processing and computer vision techniques to address changes in illumination and atmospheric conditions. Refining the model with data that looks like true world variations boosts the system's stability.

Data Diversity: Obtaining a wide range of data which has different models of vehicles, shiny and dull textures, a variety of angles, lighting conditions, and backgrounds which is significant for designing models that effectively detect vehicles. The data that is used to train the model can be very limited or even biased which can cause the model to have a poor performance on the real-world data.

Model Generalization: One of the hardest issues is to guarantee that the detection model can generalize well to cover new or unheard car types. Tendency towards certain models in the training data set may result in poor generalization performance resulting in less efficient working with different models.

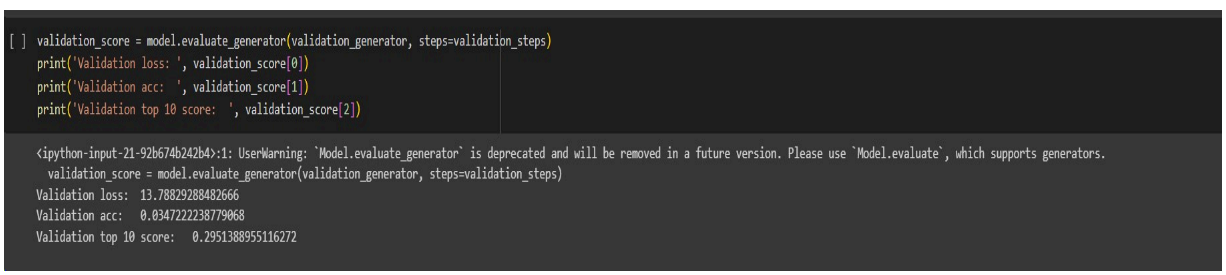
Scale and Resolution: Precisely tracing various vehicle types in different scales and resolutions is a basic requirement for the real-world applications. It is difficult to deal not only with the difference in image quality and resolution but also in some cases with the ultra-high resolution geospatial data used in the surveillance and remote sensing.

Occlusion and Clutter: Real-world scenes have foreground objects that can cause vehicles to be occluded from roadside or there is a lot of jam due to which vehicles become unclearly visible. The creation of algorithms that can properly identify the model of a vehicle even in the most cluttered and occluded situations is a real problem.

Real-time Processing: Real-time operating capability is the key factor to ensure proper functioning of traffic monitoring systems and surveillance in vehicle model detection systems. The most difficult part is how to optimize the algorithms in order to make sure that the algorithms stay fast, effective and accurate.

CHAPTER 4: TESTING

The experimental testing phase is critical for ascertaining whether MobileNetV2 performs effectively or not. The aim is to use defined tools and methods to calculate the precision and reliability of the model on the basis of various situations. To carry out a quantitative assessment of the model's performance, evaluation metrics such as precision, recall and F1 score are used. By using visualization tools such as TensorBoard, one can have clarity on the training process for instance loss functions across epochs, ensuring that the model converge. Such test cases cover the accuracy of the model on validation data aimed at checking its stability for not yet being trained data. Another aspect is checking out how well such a system operates in various ambient conditions and can serve as an assurance that it will work well even in practice. This stage tests whether the various quantitative measurements combined with some visualization provides sufficient insights on what MobileNetV2 can do, which in turn will inform further optimization where necessary.



```
[ ] validation_score = model.evaluate_generator(validation_generator, steps=validation_steps)
    print('Validation loss: ', validation_score[0])
    print('Validation acc:  ', validation_score[1])
    print('Validation top 10 score:  ', validation_score[2])

    <ipython-input-21-92b674b242b4>:1: UserWarning: `Model.evaluate_generator` is deprecated and will be removed in a future version. Please use `Model.evaluate`, which supports generators.
      validation_score = model.evaluate_generator(validation_generator, steps=validation_steps)
    Validation loss:  13.78829288482666
    Validation acc:   0.0347222238779068
    Validation top 10 score:   0.2951388955116272
```

Fig 7: Testing

CHAPTER 5: RESULT AND EVALUATION

The development of accurate and effective models for vehicle model detection and color could be used in various applications such as traffic management, road safety interventions and surveillance. Aimed at leveraging deep learning to develop an effective system for detection of vehicle type and colors from pictures.

In the course of this venture, we traveled through an intensive process that included preparation of inputs, modeling, learning, assessment, as well as forecasting. The resultant model is able to identify vehicle models and colors from pictures quite well and uses the latest approaches of deep learning technologies.

Key Achievements:

1. Data Preprocessing: Data preprocessing were carried out extensively in the commencement of the project, including collecting, augmenting and normalizing the data. In this way, the model was exposed to different data that were a true representation of reality.

2. Model Architecture: For the vehicle model and color detection model, we chose to adopt the MobileNetV2 architecture as the backbone. Having highly effective architecture and state of art parameters set a good stage for our deep learning model.

3. Training and Evaluation: A trained dataset was used in refining the model. Training included measures like early stopping, model checkpointing, and dynamic learning rate reduction in order to stop overfitting and ensure better convergence. The model's performance was assessed using evaluation metrics including loss, accuracy, and top-10 accuracy.

4. Prediction and Result Interpretation: Therefore, this trained model showed its ability in making correct predictions regarding new and unseen images. Successfully, we generated predictions using preprocessed input images after loading the trained model to interpret the data provided insights on the models of vehicles and color identification.

The script initializes after importing the required dependencies and defines a custom top-10 accuracy metric. It then loads the pre-trained model ‘best_model.h5’ and resizes an image specified by ‘img_path’ to the target size of 224x2 Afterward, it then reads a pickled file named “classes_name.pickle” that has class labels linked to a trained model. Then, the loaded model estimates class probabilities for the given picture, and the top 5 predictions are extracted based upon the highest probabilities. This results in the presentation of a table showing the predicted class labels and their respective probabilities expressed in percentages. The output of this helps us better understand our model’s trustworthiness and correctness in labeling the presented picture. To make sure the libraries are loaded, there must be the correct model and class label files placed in their relevant locations respectively.

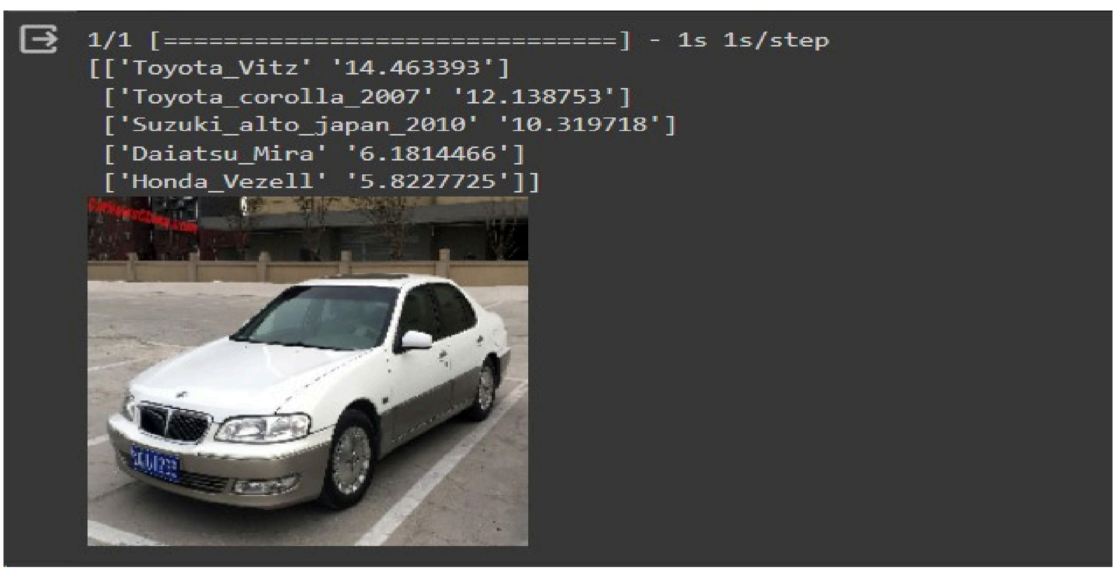


Fig 8: Result

CHAPTER6: CONCLUSIONS AND FUTURE SCOPE

6.1 CONCLUSIONS

Achievement of Objectives: Its success is evidenced through developing and deploying an intelligent vehicle model detection system that correctly distinguishes vehicle makes and colors out of various images/live videos.

Technological Advancements: This state-of-the-art model is highly effective, and it shows efficiency and accuracy in real-time vehicle detection due to the utilization of advanced deep learning techniques such as using the MobileNetV2 architecture.

Impact and Applications: This developed system features many possible uses, such as improving traffic control and monitoring, automatically performing process, transportations, safety and urban planning.

User-Centric Approach: A user has a friendly interface that makes possible easy loading of an image/a video stream for determining a car model and color, which increases a system's convenience.

Real-World Implications: This Vehicle Model Detection System promises success in the real world with the advantage of providing automated and improved services.

Continued Development and Collaboration: To ensure that the system stays relevant and effective, it is important to collaborate with industry stakeholders while continuously augmenting the data and redesigning the model.

Data Preprocessing: Gathering, augmenting, and normalizing data was part of the project's comprehensive data preprocessing phase that got underway. By ensuring that the model is trained on a variety of representative data, this stage improved the model's capacity for generalization to actual situations.

Training and Evaluation: Using the prepared dataset, the model underwent extensive training. To avoid overfitting and maximize results, early stopping, model checkpointing, and dynamic learning rate reduction strategies were included to the training process.Assessment criteria like accuracy, loss, and top-10 accuracy were employed to measure the performance of the model.

Prediction and Interpretation of Results: On fresh, unviewed photos, the trained model proved its ability to produce correct predictions. We were able to effectively create predictions and analyze the findings by loading the trained model and preprocessing the input photos. This allowed us to gain important insights into the recognised vehicle models and colors.

Enhanced Model Performance: Augmentation approaches, on the other hand, are becoming a vital factor in the performance improvement processes involved in vehicle model detection models and color identification algorithms. The model is made more robust, general, and accurate when it is trained with a vast assortment of variations.

Mitigation of Data Scarcity: This issue is solved by utilizing an augmentation technique. An additional information is introduced to the initial training data making them vaguer and more extensive. This grants models the chance to study from large collections of examples and features, regardless of the size of the training dataset.

Robustness to Environmental Variations: Augmentation ensures that the vehicle model detection models and the color identification algorithms are able to handle the variations in the environmental conditions like the changes in the lighting, weather and the occlusions.

Balancing Realism and Diversity: Unlike its opposing nature, one of the main obstacles in augmentation is the proper mixture of various variations into the initial set of images, while preserving the original look of the augmented pictures. The best augmentation methods should preserve the real-world aspect of the images and expose the model to a diversity of images for learning the robust features.

Integration with Visual Features: Through augmentation, visual cues such as vehicle model detection and color identification are better highlighted, and in turn, the models are proved to be more powerful and interpretable. Through augmented dataset which consist of visual attributes for the model, models can take advantage of the mix of semantic and spatial information for top performance.

Ethical and Legal Considerations: The moral and legal aspects are the most crucial in the development and deployment of the augmented vehicle model detection models and color identification algorithms for this application. Fairness, transparency and competence of collecting, annotating and training data must be taken into account to prevent bias and maintain ethical standards.

6.2 FUTURE SCOPE

Video Frame Extraction: Using a file or a live camera feed, extract video frames from the supplied video stream.

Frame preprocessing: Before entering a video frame into the car model detection system, make sure it is the right size and format.

Visualization: Using labels indicating the make and model and bounding boxes surrounding detected cars, visualize the detection results on the video frames.

Fine-tuning: Adding more diverse and annotated data to the model could help it become even more accurate and resilient.

Real-Time Application: The model's usefulness and impact could be increased by modifying it for real-time applications like traffic analysis or vehicle surveillance.

Multi-View and Occlusion Handling: Improving the model's capacity to identify cars in partially occluded situations and from different perspectives would improve its performance in challenging real-world environments.

Integration with IoT Devices: In contexts with constrained computational resources, quick deployment of the model may be possible by integration with edge computing and IoT devices.

To sum up, this research serves as an excellent example of how deep learning may be used to tackle challenging computer vision tasks. In addition to achieving accurate classification, we have also laid the groundwork for future developments and applications in the field of vehicle recognition and analysis by creating a vehicle model and color detection system.

It is the project that proves how deep learning is the key to the whole new era of human interaction with visual data in different industries.

REFERENCES

[1] Chirag, "VEHICLE DETECTION AND TRACKING USING MACHINE LEARNING TECHNIQUES," vol. 11, no. 4, 2023.

[2] A. Farid, F. Hussain, K. Khan, M. Shahzad, U. Khan, and Z. Mahmood, "A Fast and Accurate Real-Time Vehicle Detection Method Using Deep Learning for Unconstrained Environments," IEEE Transactions, 2023.

[3] C. Babi, Y. S. Gayatri, M. V. S. Vishnu, and Y. A. Shreyas, "Vehicle Model Prediction using Machine and Deep Learning Techniques," *Industrial Engineering Journal*, vol. 52, no. 4, April 2023.

[4] Y. Chen and Z. Li, "An Effective Approach of Vehicle Detection Using Deep Learning," 2022.

[5] S. Karungaru, D. Lyu, and K. Terada, "Vehicle Detection and Type Classification Based on CNN-SVM," *International Journal of Machine Learning and Computing*, vol. 11, no. 4, July 2021.

[6] S. Addala, "Vehicle Detection and Recognition," Dept. of CSE, Lovely Professional University, Punjab, India,2020

[7] V. Vijayaraghavan and M. Laavanya, "Vehicle Classification and Detection using Deep Learning," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 1S5, December 2019.

[8] H. Zhao, J. Fu, and Y. Jing, "Vehicle Model Identification Method Based on Convolutional Neural Network," 2019.

[9] N. S. Nemcev, "Image-based Approach for Vehicle Model Re-Identification," in Journal of Scientific and Technical Information Technologies, Mechanics and Optics (ITMO University), vol. 122, no. 4, Jun 2019.

[10] L. Suhao, J. Lin, G. Li, T. Bai, H. Wang, and Y. Pang, "Vehicle Type Detection based on Deep Learning in Traffic Scene," vol. 131, 564-572, 2018.

[11]Tensorflow documentation Retrieved from https://en.wikipedia.org/wiki/TensorFlow

[12]PyTorch documentation Retrieved from https://en.wikipedia.org/wiki/PyTorch

[13]OpenCV documentation Retrieved from https://en.wikipedia.org/wiki/OpenCV

[14]Tensorflow documentation Retrieved from https://www.run.ai/guides/tensorflow/tensorflow-serving

[15]Open Neural Network Exchange documentation Retrieved from https://en.wikipedia.org/wiki/Open_Neural_Network_Exchange

[16] Flask documentation Retrieved from https://www.analyticsvidhya.com/blog/2021/10/flask-python/#:~:text=Flask%20is%20used%20for%20developing,and%20a%20fast%20debugger%20provided.

[17]Google Colab Retrieved from https://colab.research.google.com/

[18]MobileNetV2 retrieved from https://www.mathworks.com/help/deeplearning/ref/mobilenetv2.html

[19]Tensorflow Tutorial for Python retrieved form https://www.youtube.com/watch?v=6_2hzRopPbQ

[20]OpenCV Python Tutorial retrieved from https://www.youtube.com/watch?v=qCR2Weh64h4&list=PLzMcBGfZo4-lUA8uGjeXhBUUzPYc6vZRn

[21]MobileNet Architecture Explained retrieved from https://www.youtube.com/watch?v=v2zM-DVSUkQ

[22]Building a Neural Network with PyTorch retrieved from https://www.youtube.com/watch?v=mozBidd58VQ

[23]ONNX (Open Neural Network Exchange) retrieved from https://www.youtube.com/watch?v=BtmnKZPdhSE

[24] Object Detection with TensorFlow retrieved from https://www.youtube.com/watch?v=dMh7krojLIE&list=PLpkFduNMb9uInRPXI-2CtNDq4DYXm9cUK

[25]Learn Flask for Python retrieved from https://www.youtube.com/watch?v=Z1RJmh_OqeA

[26]K. Nguyen, "Data Augmentation Analysis of Vehicle Detection in Aerial Images," Journal of Computer Science and Cybernetics, vol. 39, no. 3, pp. 291–312, 2023.

[27]H. Kumdakc, C. Ongun, and A. Temizel, "Generative Data Augmentation for Vehicle Detection in Aerial Images," arXiv:2012.04902v1 [cs.CV], Dec. 2020

[28]V. Sowmya and R. Radha, "Real-time Vehicle Detection implementing Deep Convolutional Neural Network features Data Augmentation Technique," ,2022.

Kaushik_Report

ORIGINALITY REPORT

6%	5%	2%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ir.juit.ac.in:8080 Internet Source	2%
2	www.ir.juit.ac.in:8080 Internet Source	2%
3	"Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications", Springer Science and Business Media LLC, 2019 Publication	<1%
4	www.ascl.net Internet Source	<1%
5	pdfslide.tips Internet Source	<1%
6	Raghad K. Mohammed, Azmi Tawfeq Hussein Alrawi, Ali Jbaeer Dawood. "U-Net for genomic sequencing: A novel approach to DNA sequence classification", Alexandria Engineering Journal, 2024 Publication	<1%
7	www.ijraset.com	

	Internet Source	<1 %
8	www.researchgate.net Internet Source	<1 %
9	www.mdpi.com Internet Source	<1 %
10	adoc.pub Internet Source	<1 %
11	rstudio-pubs-static.s3.amazonaws.com Internet Source	<1 %
12	Rômulo Teixeira Rodrigues. "Aterrissagem autônoma baseada em visão para mini quadrirrotor", Universidade de São Paulo. Agência de Bibliotecas e Coleções Digitais, 2017 Publication	<1 %
13	psaspb.upm.edu.my Internet Source	<1 %
14	Li Suhao, Lin Jinzhao, Li Guoquan, Bai Tong, Wang Huiqian, Pang Yu. "Vehicle type detection based on deep learning in traffic scene", Procedia Computer Science, 2018 Publication	<1 %
15	cloudblogs.microsoft.com Internet Source	<1 %

16	dspace.dtu.ac.in:8080 Internet Source	<1 %
17	repo.lib.tokushima-u.ac.jp Internet Source	<1 %
18	www.geeksforgeeks.org Internet Source	<1 %
19	Nermeen Nader, Fatma EL-Zahraa A. EL-Gamal, Mohammed Elmogy. "Enhanced kinship verification analysis based on color and texture handcrafted techniques", The Visual Computer, 2023 Publication	<1 %
20	open.metu.edu.tr Internet Source	<1 %
21	aclanthology.org Internet Source	<1 %
22	ijarcce.com Internet Source	<1 %
23	journal.unnes.ac.id Internet Source	<1 %
24	ntv.ifmo.ru Internet Source	<1 %
25	scholars.unh.edu Internet Source	<1 %

26 www.coursehero.com
Internet Source
<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ....

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: ______________________________ Department: __________________ Enrolment No __________

Contact No. ______________________________ E-mail. ____

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): __________________________

____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at.................... (%). Therefore, we

are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com