Automobile odometer fraud prevention with the implementation of blockchain and deep learning

Submitted By

Intisar Islam Orony

180011118

Ishtiak Ahmed Khan

180011138

Supervised By

Dr. A.R.M. Harunur Rashid

A Thesis submitted in partial fulfillment of the requirement for the degree of Bachelor of Science in Mechanical Engineering

Department of Mechanical and Production Engineering (MPE)

Islamic University of Technology (IUT)

December, 2022

Candidate's Declaration

This is to certify that the work presented in this thesis, titled, "Automobile odometer fraud prevention with the implementation of blockchain and deep learning", is the outcome of the investigation and research carried out by me under the supervision of DR. A.R.M. HARUNUR RASHID, PROFESSOR, MPE DEPT., IUT, BOARD BAZAR, GAZIPUR-1704, BANGLADESH.

It is also declared that neither this thesis nor any part of it has been submitted elsewhere for the award of any degree or diploma.

Name of the Student: Intisar Islam Orony

Student No: 180011118

Name of the Student: Ishtiak Ahmed Khan

Student No: 180011138

RECOMMENDATION OF THE BOARD OF SUPERVISORS

The thesis titled "Automobile odometer fraud prevention with the implementation of blockchain and deep learning" submitted by Intisar Islam Orony, Student No: 180011118 & Ishtiak Ahmed Khan, Student No: 180011138 has been accepted as satisfactory in partial fulfillment of the requirements for the degree of B Sc. in Mechanical Engineering on **19th May, 2023.**

BOARD OF EXAMINERS

1. --------------------------------------

   Dr. A.R.M. Harunur Rashid (Supervisor)

   Professor

   MPE Dept., IUT, Board Bazar, Gazipur-1704, Bangladesh.

Acknowledgment

I would like to begin by saying Alhamdullilah and grateful to Almighty Allah SWT who made it possible for me to finish this project successfully on time. I would like say a big thank to my supervisor Dr. Abu Raihan Md. Harunur Rashid for all his support, ideas about experiments, discussions, time and for explaining so patiently the topics and checking this thesis, above all his care and concern. These will ever remain in my memory. Thanks to my examiners for their constructive ideas, suggestions and checking my work.

Special thanks to all faculties for their support and encouragement throughout the period of my studies.

Finally, I would like to express my lasting gratitude to my family members for their patience, support, encouragement, enthusiasm and prayers.

Abstract

Automobile odometer fraud prevention has been an issue for the automobile industry for some time. The estimated annual financial damages from this fraud exceed one billion dollars. It is necessary to have a solution for vehicle data that is both secure and immutable. In response to the requirements, we have chosen to implement Blockchain technology to combat automotive odometer fraud. In our study, we demonstrate and describe a comprehensive fraud prevention solution based on Deep Learning & Ethereum Blockchain technology. Previously, there have been research that used blockchain to prevent odometer fraud. However, all of these systems had one flaw that may jeopardize the system's security even before integrating blockchain. The OBD2 port, which is utilized to obtain the odometer reading, necessitates the presence of a physical adapter in the vehicle. This raises major security concerns since any tampering with the adaptor would result in odometer data alteration even before the data is deployed on the blockchain. As a result, we propose a novel solution that addresses this issue. We used state-of-the art object detection models based on CNNs to extract the odometer reading from the image and cross-validate it with the odometer reading from the adapter. The odometer reading is then uploaded to the blockchain leveraging smart contracts. We developed a comprehensive system architecture to prevent odometer fraud and addressed security risks associated with OBD2 adapters used in the process of extracting odometer readings.

Table of Contents

List of Figures

List of Tables

Nomenclatures

Abbreviation	**Meaning**
ETH	Ethereum Currency
HEX	Hexadecimal Value
PoW	Proof of Work
PoS	Proof of Stake
EVM	Ethereum Virtual Machine
ERC20	Ethereum Request for Comment 20
GPS	Global Positioning System
IoT	Internet of Things
OBD	On-board Diagnostics
ABI	Application Binary Interface
ECU	Electronic Control Unit
USB	Universal Serial Bus
JSON	JavaScript Object Notation
PID	Parameter IDs
IDE	Integrated Development Environment

Chapter 1: Introduction

In this introduction section, the background and context of the topic being investigated are provided. The research problem and objectives are clearly defined and the significance of the study is explained. A brief overview of the research methods used is also included. The introduction serves as an overview of the entire research paper and provides the reader with a general understanding of the purpose and scope of the study. It also sets the stage for the rest of the paper by introducing the key concepts and research questions.

1.1 Background of the Study

1.1.1 Deep learning Technology

Deep Learning (DL) is a field that has made a major impact on data science in the last decade. It involves designing deep neural networks using basic structures and applying them to popular use cases. Neural Networks (NN) have revolutionized modern life and their impact is evident in many fields such as medical imaging, text-to-speech, natural language processing, optics, image processing, and computer vision. The development of NN began in the mid-1960s with the publication of the Perceptron and has since undergone several advancements and decelerations. The introduction of the multi-layer perceptron and the backpropagation algorithm in the mid-1980s marked a significant development in the field. Today, almost every research field has been affected by NN, experiencing significant improvements in abilities and performance. The key enablers for the current success of NN are the large amounts of data available today and the developments in GPU computations that accelerate training time significantly.

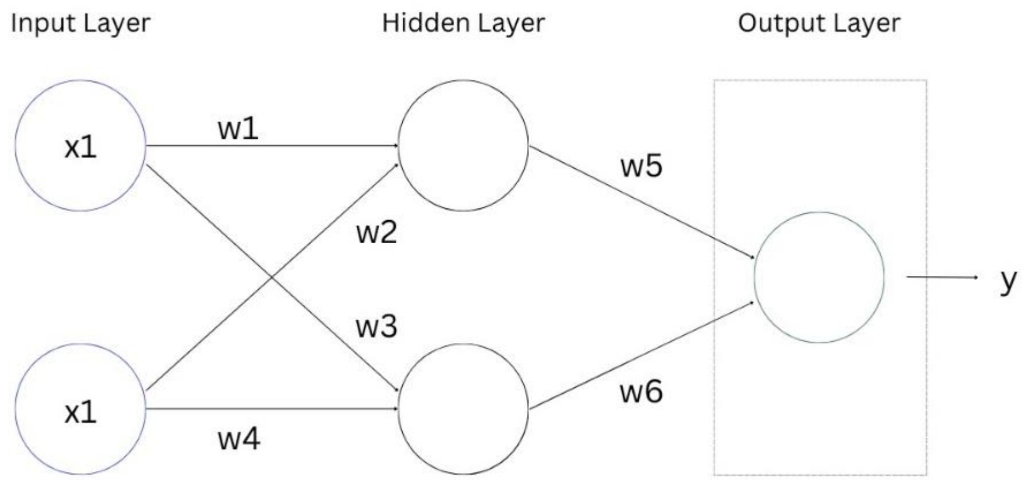


Figure 1 Deep Learning Neural Network

the basic building block of a NN consists of a linear operation followed by a non-linear function. Each building block consists of a set of parameters, termed weights and biases, that are updated in the training process with the goal of minimizing a pre-defined loss function. A common basic NN building block is the Fully Connected (FC) layer, where every neuron in one layer is connected to every neuron in the following layer. Another common layer is the convolutional layer, which applies one or multiple convolution filters to its input. Common non-linear functions applied element-wise are known as activation functions and include Rectified Linear Unit (ReLU), leaky ReLU, Exponential Linear Unit (ELU), hyperbolic tangent (tanh), and sigmoid. [1]

1.1.2 Blockchain Technology

Blockchain is a novel data storage solution that uses a distributed, smart contract which stakeholders use to confirm and indelibly record exchanges over data networks. Distributed nodes obtain agreement on groups of transactions by voting to validate the correct database version, so ensuring the permanence and integrity of information with the combination of immutability and trustless data consolidation. Organizing the series of payments into segments which are linked in line with predetermined principles. Each block is generated in a decentralized way at predetermined intervals using a consensus process, and the latest iteration of the blockchain is recorded by the nodes on the network. The module includes an identifier, a reference towards the hash of the preceding block, and data. As illustrated in Figure 1, it holds its own signature, which is generated based on all of the stated attributes. [2]

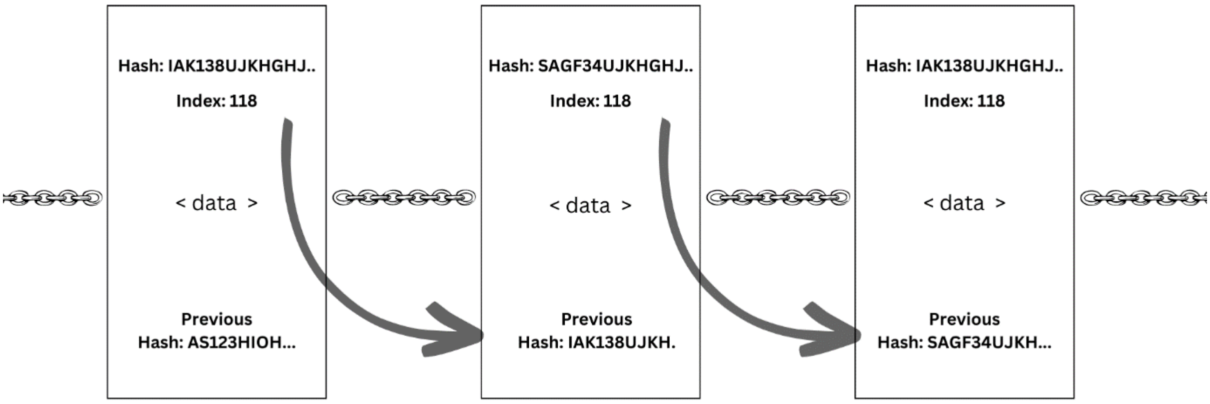


Figure 2: Blockchain Architecture

1.1.3 History of Blockchain Technology

In 2008, Satoshi Nakamoto introduced Bitcoin which is the first application of blockchain. Blockchain is the technology powering Bitcoin and many other cryptocurrencies; it aims to address two types of issues: generating an indelible ledger of statistical information and preserving unanimity in a secured network. Using a blockchain, cryptocurrencies are financial apps that may be created. [3]

1.1.4 Blockchain in Automobile Industry

Blockchain has garnered significant interest from business, academia, and the government. It has the ability to change a number of application domains and is regarded as a rapidly emerging force in technology. The automotive sector could be revolutionized by blockchain technology.

Several information pertaining to automotive components and functionality is essential to the growth of the automobile sector. The odometer of a vehicle, for instance, is a way of monitoring a vehicle's milage. This device is easily susceptible to tampering, resulting in significant losses for automobile purchasers and linked companies, while detecting fraud can be extremely difficult.[3]

1.1.5 Odometer Fraud Situation

Recent estimates by the Parliament of Europe sets that the annual financial cost of mileage manipulation within their countries is between 4 and 10 billion euros. [4]

1.1.6 Odometer Fraud Detection in Current Ways

Presently, physical inspection and paperwork verification are the primary methods used to identify mileage manipulation. But, neither of these methods are clear evidence of mileage manipulation; there could have several manipulations or have been carefully explained by the scammers. In addition, there are no preventive interventions available, only retrospective analysis.[3]

1.1.7 Odometer Fraud Detection using Blockchain Technology

Since blockchain data is unchangeable, odometer manipulation might be quickly identified and rendered ineffective by storing mileage information in real-time on the blockchain. The following are the primary contributions of this article in this context:

• It addresses information entry, dispersed authorization of data structure, routing algorithms and components, as well as smart contracts in a way that has never been done before for the automobile sector and blockchain design.[3]

• It presents and explores the applicability of blockchain technology in routing protocol, comparing previous studies conducted.[3]

• This proposes a revolutionary blockchain based solution for the irreversible registry of a car's mileage readings, accounting for a networked consortium of several stakeholders.[3]

1.2 Scope and Novelty

Our approach to preventing odometer fraud is designed to be applicable to all contemporary automobiles, regardless of make or model. This universality stems from the fact that virtually all new vehicles are equipped with an On-Board Diagnostics 2 (OBD2) port. Leveraging this standardized feature, our fraud prevention system can be implemented across the majority of vehicles currently available on the market.

Previous studies have explored the use of blockchain technology for preventing odometer fraud. However, these systems typically rely on an onboard OBD2 adapter for acquiring the odometer reading, which introduces vulnerabilities to tampering. This compromises the integrity of storing the odometer data onto a decentralized blockchain system, as the initial data collection process is susceptible to manipulation.

In order to address this challenge, we propose a novel approach that leverages deep learning models to analyze automobile dashboard images. Our model is trained to precisely localize the region of interest within the image, which contains the desired odometer reading. Subsequently, another deep learning model based on Optical Character Recognition (OCR) techniques is utilized to accurately extract the numeric information from the localized region. In our proposed system, the extracted odometer reading acts as a validation mechanism for the data obtained from the OBD2 USB adapter. By utilizing this approach, we can ensure that the odometer reading is obtained with a high level of accuracy and eliminates the risk of tampering associated with the OBD2 adapter. The validated reading is then securely stored within the blockchain using smart contracts, ensuring the immutability and transparency of the recorded data.

This methodology enables us to overcome the limitations of relying solely on the OBD2 adapter by incorporating advanced deep learning techniques for robust data extraction. The resulting system offers a comprehensive solution to prevent odometer fraud, ensuring the integrity and reliability of recorded odometer readings within the blockchain ecosystem.

1.3 Objectives with specific aims

- Develop a tamper-proof system using blockchain to record and store vehicle odometer readings.

Aim: Prevent fraudulent odometer tampering and increase transparency in the automotive industry

- Implement a deep learning model for accurate detection of odometer data.

Aim: Improve efficiency and effectiveness in identifying manipulated odometer readings using machine learning.

- Explore the feasibility of integrating blockchain and deep learning for real-time odometer fraud detection.

Aim: Provide a reliable solution to combat the rising issue of odometer fraud in the automotive ecosystem.

- Evaluate the financial impact of blockchain and deep learning in reducing odometer fraud-related losses.

Aim: Quantify cost savings and benefits from implementing the proposed system and technologies.

- Ensure the security and privacy of sensitive data in the blockchain-based fraud prevention system.

Aim: Establish a secure infrastructure while preserving the privacy of vehicle owners and users.

- Assess scalability and performance for handling large volumes of vehicle data and real-time fraud detection.

Aim: Maintain fast and accurate fraud detection as the system scales with the increasing number of vehicles.

- Evaluate different blockchain and deep learning architectures for odometer fraud prevention.

Aim: Identify the most effective combination of technologies for fraud prevention in the automotive industry.

1.4 Structure of the Thesis

1.4.1 Convolutional Neural Networks

Artificial Neural Networks (ANNs) are computational processing systems that are heavily inspired biological nervous systems, such as the human brain, operate. ANNs are mainly comprised of a high number of interconnected computational nodes (referred to as neurons), which work in a distributed fashion to collectively learn from the input in order to optimize its final output.

Convolutional Neural Networks (CNNs) are analogous to traditional ANNs in that they are comprised of neurons that self-optimize through learning. Each neuron will still receive an input and perform an operation (such as a scalar product followed by a non-linear function). From the input raw image vectors to the final output of the class score, the entire network will still express a single perceptive score function (the weight). The last layer will contain loss functions associated with the classes, and all of the regular tips and tricks developed for traditional ANNs still apply.

The only notable difference between CNNs and traditional ANNs is that CNNs are primarily used in the field of pattern recognition within images. This allows us to encode image-specific features into the architecture, making the network more suited for image-focused tasks while further reducing the parameters required to set up the model. The Object Detection architecture we are using are based on CNNs. [5]

1.4.2 Transfer Learning

Transfer learning is a machine learning technique that aims to improve the performance of a model on a target task by transferring knowledge from a related source task. This approach can reduce the dependence on substantial amounts of labeled data for the target task, as it leverages knowledge from the source task to improve the model’s performance. Transfer learning has become a popular and promising area in machine learning due to its wide application prospects. It has been used in various fields such as computer vision, natural language processing, and speech recognition. Transfer learning can be divided into several categories based on different criteria, including the type of transfer and the relationship between the source and target domains. In our systems we leverage the Transfer Learnings to help our model converge and find the global minimum. [6]

1.4.3 Ethereum

The Ethereum blockchain is similar to the Bitcoin blockchain. In addition to the identifier, time stamp, etc., blocks of Ethereum also contain a transaction record and the input data. The new state is constructed by applying the prior state to each contract in the exchange record. Ethash, Ethereum's PoW algorithm, is memory intensive and represents the Dagger-Hashimoto algorithm's modification. Every node in the Ethereum protocol executes the Ethereum Virtual Machine and its associated commands. The written smart contracts are transferred into the

EVM so that nodes can read and execute them. Solidity is among the most popular programming languages for creating smart contracts. The typical Ethereum transaction duration is around 10 to 20 seconds, sometimes it rises up to 35 seconds. [7]–[9]

1.4.4 Transactions through Ethereum and ERC20 Token

A transaction is a unique command that has been cryptographically signed. This transaction is a validated data file sent from an external user. Each transfer includes the message destination, the sender's hash, the Ether amount, an additional input section, and the STARTGAS and GASPRICE. Ethereum as a network is suitable for its issuance of tokens. Smart contracts are Ethereum-based contracts that practice the ERC20 Token Standard. [7]–[11]

1.4.5 Design Choices

The selection of a consensus algorithm is crucial to the development of an effective blockchain. The most essential considerations are security, performance, and decentralization Noting that upgrading the consensus method in an active blockchain is an extremely hard operation that typically results in a split network where several nodes continue to run outdated code meanwhile others adapt to the new approach, it is crucial to choose the suitable design. In many circumstances, a private blockchain is preferable. Therefore, we chose a private blockchain network with PoW consensus on the Ethereum blockchain network for our use case. [3]

Chapter 2: Literature Review

In this literature review section, a comprehensive examination of the existing literature on the topic was conducted. A thorough search of academic databases and journals was performed to gather relevant studies and articles. The findings of these studies were analyzed and synthesized to provide a clear understanding of the current state of research on the topic. The literature review also highlights the key gaps and areas that still require further investigation. This section serves as a foundation for the current study and provides a basis for the research questions and hypotheses. A summary of the literatures reviewed is given in Table 1.

Table 1: Literature Review

Sl no.	Year	Input	Limitations	Reference
1	2017	Created the first-ever prototype of a cloud-based manual for auto maintenance.	All of the projects violate the decentralized architecture of blockchain networks and are in their early stage.	[12]
2	2019	Present a blockchain-based system for managing and storing distributed automobile data.	Managed by a single organization and in its infancy stage.	[13]
3	2019	A ledger system that provides car information	Omits the repair history of the vehicle. Contrary to the decentralized design of Blockchain	[14]

		including ownership, mileage, manufacturer, and recovery record.	technology, it would be based on a single entity running a Blockchain application. For certification, a collaborative environment must be established, which does not also permit information access privacy and security.	
4	2017	Present a method for combating odometer fraud that employs blockchain technology to ensure anonymity. Using a dongle, the program logs the car's GPS and distance information and stores it on the Ethereum Blockchain.	The security and maintenance of the cloud database used are not explored in respect to key management and distribution issues. With access to a user's unique identifier, anyone can edit transaction records and invalidate storage, posing security problems. The Ethereum platform has scalability and cost difficulties, the information flow between client and program is open and susceptible to tampering hazards, and the architecture as a whole is unsafe and requires security checks.	[15]
5	2019	It is necessary to address the issue of	There is a limit on the registration of assets. Customer licenses do not	[16]

		falsification in the reporting of tangible documents, such as automobile mileage records.	address corruption and loss issues. The client app, the customer's credit interaction, as well as other operations are not covered in the paper. Additionally, neither the proposed approach of security risks was evaluated nor were privacy concerns mentioned. This study of communication is so elementary that it was unable to detect the owner's stated false readings.	
6	2018	Develop a Dual Data Encryption Framework for the Consortium Blockchain in order to facilitate the exchange of automobile information.	Issues pertaining to user identification, identity management, and cost-sharing mechanisms remain unresolved.	[17]
7	2020	Utilizing smart contracts on the Ethereum Network and the OpenXC IoT gateway, a strategy	Privacy concerns are not considered. There was no mention of user identification or permission requirements. When data is simply collected by sensors, the roles of	[18]

		for protecting vehicle data was proposed.	stakeholders are not taken into account. It is not a comprehensive solution for safely managing vehicle data.	
8	2020	A proof-of-concept model for the exchange of tangible goods using smart contracts is given. Additionally, the prototype provides an ownership history of the products.	It makes no mention of how to get and maintain accurate information regarding a vehicle's features and service records.	[19], [20]
9	2021	A proposed methodology for securing vehicular networks using blockchain and restricting third party requests.	It only identifies the flaw and proposes an architecture framework but doesn't provide the knowledge that if the idea can be practically applied.	[21]
10	2018	A blockchain backed vehicle data framework is suggested to address the issue of	The framework has not been tested practically, and the challenges for implementing it are not explained properly.	[22]

		automobile odometer fraud and also addresses the lack of unified vehicle life-cycle management.		
11	2015	A blockchain-based sensor data security system for IoT devices has been proposed to ensure data security and privacy using data authentication.	The design principles described are based on a particular theoretical lens, other lenses may specify different or new design requirements and principles Furthermore, the evaluation of the model is limited to three applications, and may require experimentation others to highlight the design at various levels.	[23]
12	2020	It focuses on blockchain architectures and techniques for managing large numbers of IoT devices and enabling smart city infrastructure.	It provides only a summary of each case, without delving into the specifics or implications of the research.	[24]

13	2021	It introduced the need to apply blockchain technology in the used car market to reduce fraud and increase trust by recording vehicle lifecycle events in a secure ledger.	The findings do not include discussions about stakeholder incentives for blockchain management and data privacy.	[25]
14	2019	The study presents a framework for evaluating and classifying blockchain use cases in the mobility domain and provides an overview of promising use cases and their challenges.	The study does not provide a comprehensive assessment of the technical, economic and legal challenges of implementing blockchain technology in the mobility sector.	[26]
15	2020	The study proposes the use of distributed hash tables (DHT) and public blockchains to	The effect of this architecture in the long term is not explained properly.	[27]

		provide a high-assurance, scalable and efficient solution to ensure data leaks to ensure the integrity of vehicle software and they are reliable.		
16	2012	The PASCAL Visual Object Classes (VOC) challenge is a benchmark in visual object category recognition and detection		[28]
17	2020	The Optical Character Recognition (OCR) systems have been widely used in various of application scenarios, such as office automation (OA) systems,	However, OCR is still a challenging task due to the various of text appearances and the demand of computational efficiency	[29]

		factory automations, online educations, map productions etc		
18	2019	In object detection, key point-based approaches often suffer a large number of incorrect object bounding boxes, arguably due to the lack of an additional look into the cropped regions	CenterNet achieves an AP of 47.0%, which outperforms all existing one-stage detectors by at least 4.9%. Meanwhile, with a faster inference speed, CenterNet demonstrates quite comparable performance to the top-ranked two-stage detectors.	[30]
19	2015	It represents a method for detecting objects in images using a single deep neural network.	Compared to other single stage methods, SSD has much better accuracy, even with a smaller input image size. For 300×300 input, SSD achieves 72.1% mAP on VOC2007 test at 58 FPS on a Nvidia Titan X and for 500×500 input, SSD achieves 75.1% mAP, outperforming a comparable state of the art Faster R-CNN model	[31]

20	2015	State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. Advances like SPPnet and Fast R-CNN have reduced the running time of these detection networks, exposing region proposal computation as a bottleneck.	while achieving state-of-the-art object detection accuracy on PASCAL VOC 2007, 2012, and MS COCO datasets with only 300 proposals per image. In ILSVRC and COCO 2015 competitions, Faster R-CNN and RPN are the foundations of the 1st-place winning entries in several tracks.	[32]

Chapter 3: Description of the system

In section 3 and 4, the methods and procedures used to conduct the study are detailed. The research design, participants, data collection and analysis techniques are all described in this section. The chosen methodology is appropriate for the research question and helps to ensure that the data collected is valid and reliable. This section provides a clear and transparent account of the research process, allowing readers to understand and evaluate the findings of the study.

3.1 Localization of odometer region (RoI) using Object detection models

The localization of the odometer region, also known as the Region of Interest (RoI), plays a crucial role in our proposed system for preventing odometer fraud. To achieve accurate and efficient localization, we leverage state-of-the-art object detection models, which are well-established in computer vision and deep learning domains.

Object detection models excel in identifying and localizing specific objects within an image. In our case, the objective is to precisely locate the region of the dashboard image that contains the odometer display. This task poses a significant challenge due to variations in vehicle dashboard designs, varying sizes and shapes of odometer displays, and potential occlusions or interferences caused by other dashboard elements.

To address this challenge, we employ advanced object detection models, such as Single Shot MultiBox Detector (SSD) and Faster R-CNN (Region-based Convolutional Neural Network), which have demonstrated exceptional performance in object localization tasks. These models utilize deep neural networks with carefully designed architectures to efficiently detect and localize objects of interest.

During the training phase, the object detection models are fed with annotated dataset comprising dashboard images and corresponding bounding box annotations of the odometer region. This process allows the models to learn and generalize the characteristics of the odometer display, enabling them to accurately identify its location within new, unseen images.

Once the models are trained, they can be applied to novel dashboard images in real-time. By analyzing the input image, the models generate bounding box predictions that tightly enclose the identified odometer region. These bounding boxes provide spatial coordinates, allowing for precise localization of the RoI.

3.2 Architecture Used

The Object detection models are trained to localize the RoI and successfully classify it. The Object Detection Architectures used are:

- Faster R-CNN with Region Proposal Networks (RPN)
- SSD: Single Shot MultiBox Detector
- CenterNet: Keypoint Triplets for Object Detection
- Deep Residual Learning for Image Recognition

3.2.1 Faster R-CNN with Region Proposal Networks (RPN)

Faster R-CNN is an object detection framework that builds on the previous R-CNN and Fast R-CNN models. It consists of two main components: a Region Proposal Network (RPN) and a Fast R-CNN detector. The RPN is a fully convolutional network that proposes potential object regions, while the Fast R-CNN detector uses these proposals to classify the objects and refine their locations.

The RPN takes an image as input and outputs a set of rectangular object proposals, each with an objectness score. It does this by sliding a small network over the convolutional feature map output by the last shared convolutional layer. At each sliding-window location, the network simultaneously predicts multiple region proposals and their objectness scores. These proposals are then refined using non-maximum suppression to reduce redundancy.

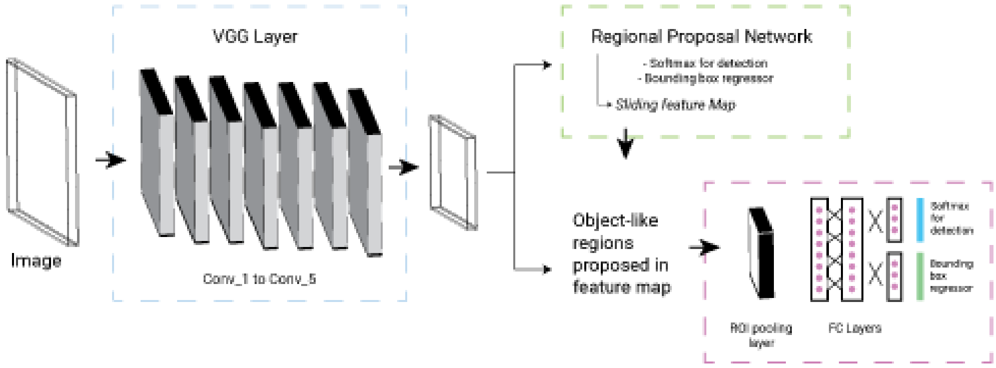


Figure 3 Faster RCNN Architecture

The Fast R-CNN detector takes the proposed regions as input and uses a RoI pooling layer to extract a fixed-length feature vector from the feature map for each proposal. These feature vectors are then fed into a sequence of fully connected layers to produce class scores and bounding box refinements.

Faster R-CNN can be trained end-to-end by alternating between fine-tuning the RPN and fine-tuning the Fast R-CNN detector while keeping the proposals fixed. This results in a unified network with shared convolutional layers that can efficiently perform object detection. [32]

3.2.2 SSD: Single Shot MultiBox Detector

SSD (Single Shot MultiBox Detector) is an object detection framework that aims to provide a balance between speed and accuracy. It uses a single deep neural network to predict both the bounding boxes and the class probabilities for multiple objects in an image.

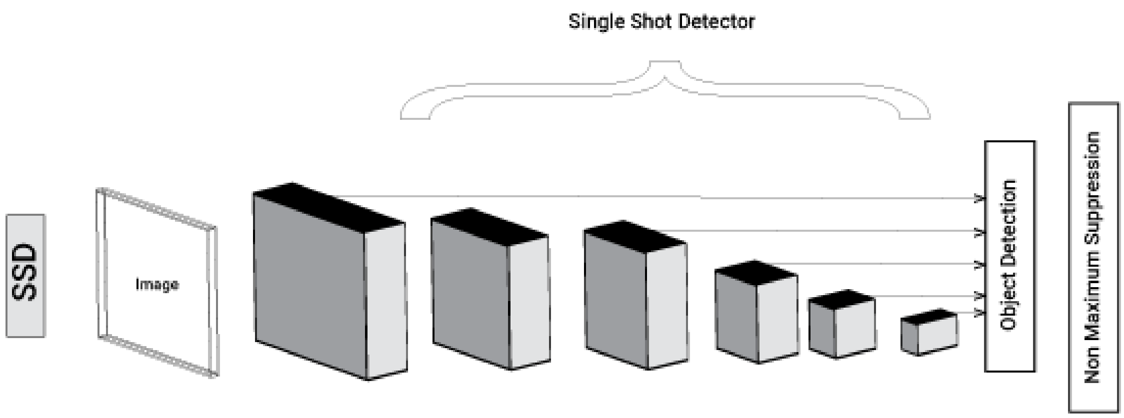


Figure 4 Architecture of Single Shot Detector

SSD works by dividing the input image into a grid of cells and predicting a fixed number of bounding boxes and class probabilities for each cell. These predictions are made at multiple scales using feature maps from different layers of the network, allowing the model to detect objects of various sizes.

The model is trained using a multi-task loss that combines localization loss (for bounding box regression) and confidence loss (for class probability prediction). During inference, the predicted bounding boxes are refined using non-maximum suppression to reduce redundancy.

In summary, SSD is an object detection framework that uses a single deep neural network to predict both the bounding boxes and class probabilities for multiple objects in an image at multiple scales. It is trained using a multi-task loss and uses non-maximum suppression during inference to refine the predicted bounding boxes. [31]

3.2.3 CenterNet: Keypoint Triplets for Object Detection

CenterNet is an object detection framework that builds upon a keypoint-based detector named CornerNet. It improves both precision and recall by detecting each object as a triplet, rather than a pair, of keypoints. This is achieved through the use of two customized modules: cascade corner pooling and center pooling. Cascade corner pooling enriches the information collected by both top-left and bottom-right corners, while center pooling provides more recognizable information at the central regions. On the MS-COCO dataset, CenterNet achieves an AP of 47.0%, outperforming all existing one-stage detectors by at least 4.9%. Additionally, with a faster inference speed, CenterNet demonstrates comparable performance to top-ranked two-stage detectors.[30]

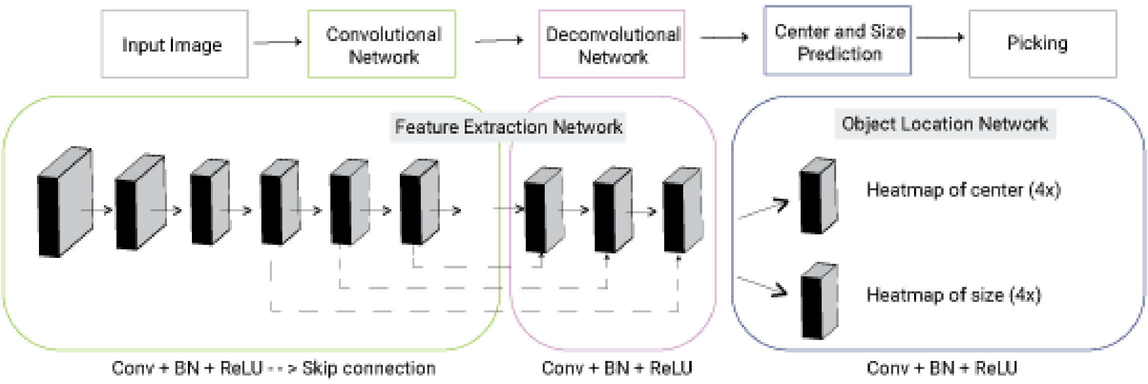


Figure 5 Architecture of Centernet Detector

3.2.4 PP-OCR model

the PP-OCR system consists of three parts: text detection, detection boxes rectification, and text recognition. The text detection part uses Differentiable Binarization (DB) as a text detector based on a simple segmentation network. The detection boxes rectification part transforms the text box into a horizontal rectangle box for subsequent text recognition, and a classifier is used to determine the text direction. If a box is determined to be reversed, further flipping is required. The text recognition part uses CRNN (Connectionist Temporal Classification) as a text recognizer, which integrates feature extraction and sequence modeling. To enhance the model ability and reduce the model size of a text recognizer, several strategies are used, including light backbone, data augmentation, cosine learning rate decay, feature map resolution, regularization parameters, learning rate warm-up, light head, pre-trained model, and PACT quantization. [29]

3.3 Dataset Annotations:

As we are using object detection architectures, we need annotated images. Object detection architectures conduct classification and localization simultaneously. So, following the Pascal VOC[28] format our dataset has been annotated. Every image contains an XML file associated with it. Below is an example of how our dataset have been annotated.

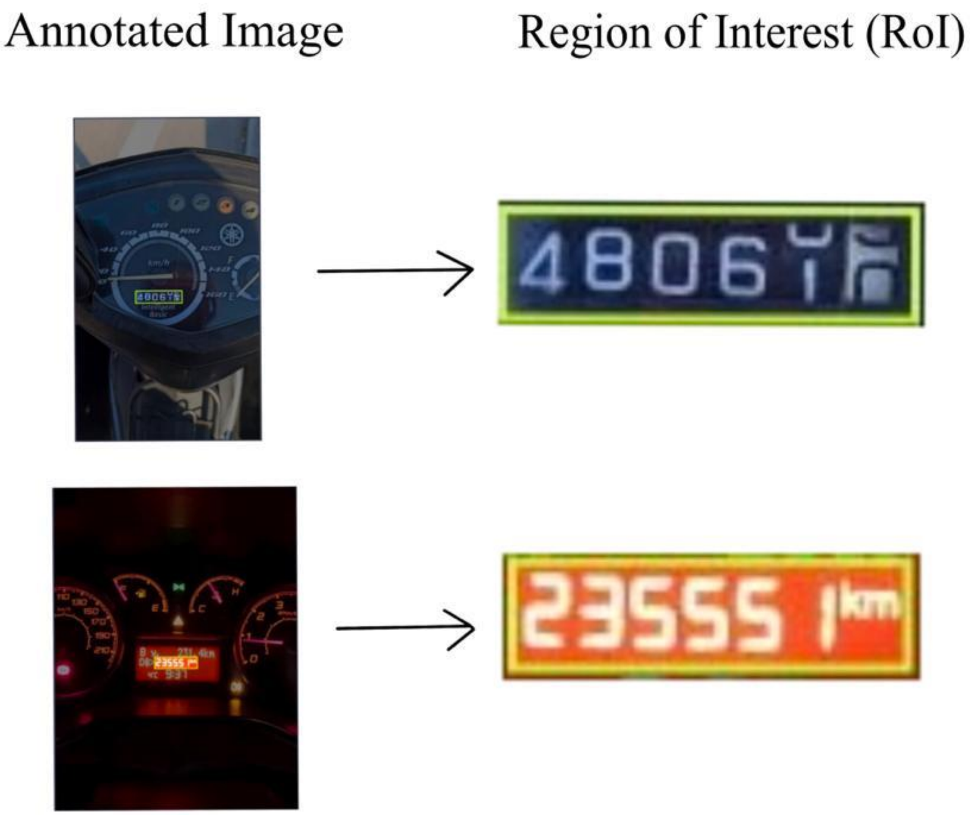


Figure 6 Identification of RoI through Annotated Image

3.4 Data Preprocessing & Augmentation:

Our initial dataset contained 2,132 images of automobile dashboard. We split the into 80/20 ratio. 80% of dataset are allocated to training set and 20% of the dataset are for evaluation. With Data Augmentation techniques the 80% training set which consists of 1,777 images is transformed into a training set of 4,300 images. The evaluation dataset is used to evaluate the models after the training has been completed.

3.5 Data Acquisition through OBD2 port of automobile

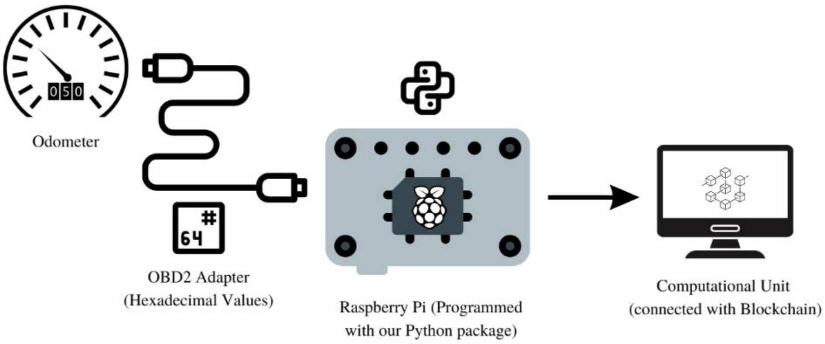


Figure 7: Data Acquisition System

Since 1996, OBD, which stands for On-Board Diagnostics, has been compulsory in the United States. Compliant OBD-2 vehicles will include a data port. It has sixteen pin locations. pyOBD is a publicly available Python scan tool that is SAE-J1979 (OBD-II) compatible. It will communicate with the car's ECU, show faulty protocols, display calculated data, etc. The application will link via the OBD-II interface, present the readings accessible for the particular vehicle, and show real time engine info in an interactive graphical user interface on the aftermarket head unit. Before beginning, a working Raspbian must be installed with network connectivity.

3.6 Smart Contract Deploy System

Smart contracts are self-executing contracts with the terms of the agreement written directly into code. They are implemented on blockchain networks, such as Ethereum, and can be used to facilitate, verify, and enforce the negotiation or performance of a contract. Using Solidity, a programming language a smart contract is written on the Ethereum blockchain.

Step 1: Writing the Contract

The first step in creating a smart contract is to write the contract code. Solidity is a high-level programming language that is similar to JavaScript and is used to write smart contracts for the Ethereum blockchain.

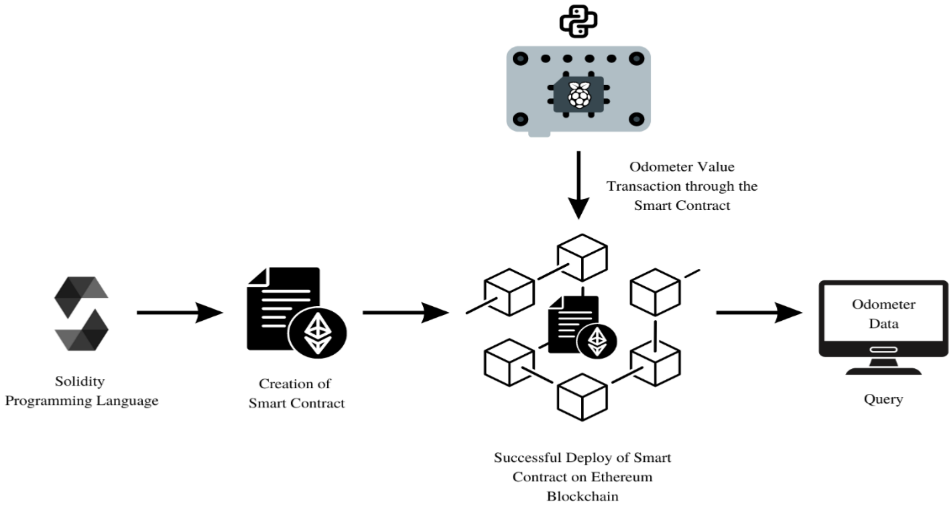


Figure 8: Smart Contract Deploy System

Step 2: Compile the Contract

Once the contract is written, we compiled it using a Solidity compiler. The most popular compiler is the Solidity compiler, which can be used to generate the bytecode required to deploy the contract on the Ethereum network.

Step 3: Deploy the Contract

We used Truffle, which is a development framework for Ethereum for this purpose.

Step 4: Interacting with the Contract

Once the contract is deployed on the Ethereum network, it can be interacted with by anyone who has the contract address and the ABI. The ABI is a JSON representation of the contract's interface, and it is required to interact with the contract. web3.js, a JavaScript library is used for interacting with the Ethereum network, to send transactions to the contract and query its state.

Chapter 4: Methodology

In this methodology section, the specific techniques and methods used for building a deep learning model, data acquisition process and smart contract deployment are discussed in detail.

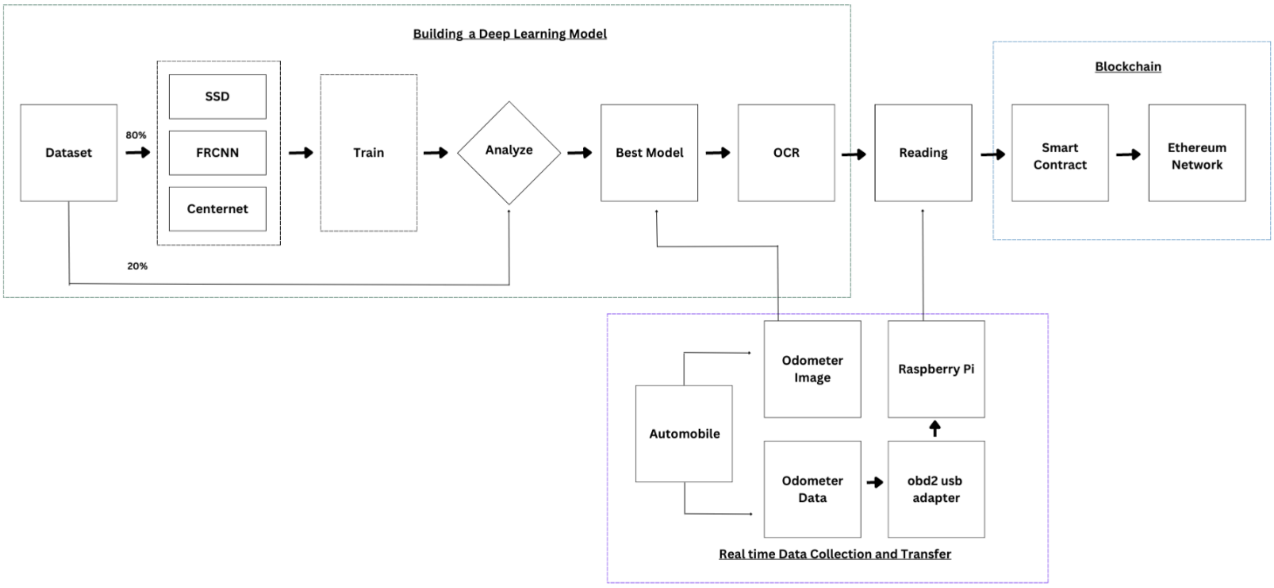


Figure 9 Architecture of proposed model

4.1 Building a Deep Learning Model

4.1.1 Dataset Collection and Preparation

A comprehensive dataset of odometer images was collected from diverse sources, encompassing various vehicle types, image qualities, and illumination conditions. Each image in the dataset was carefully annotated with ground truth values, providing accurate numeric information for evaluation. Preprocessing techniques, including image resizing, normalization, and augmentation, were applied to enhance the dataset's generalization and robustness.

4.1.2 Model Alternatives

Two state-of-the-art deep learning models, namely SSD Faster RCNN and Centernet, were chosen for training the dataset. These models are renowned for their exceptional object detection capabilities and have demonstrated impressive performance across a wide range of computer vision tasks. Transfer learning was employed to fine-tune these models, utilizing pre-trained networks such as ResNet or MobileNet as the backbone. The training process involved optimizing model parameters using stochastic gradient descent (SGD) with adaptive learning rate algorithms such as Adam or RMSprop.

4.1.3 Model Training

The training phase consisted of iteratively feeding the prepared dataset into the selected models. During this process, the models were trained to recognize and localize the digits on the odometer accurately. The objective was to optimize the models' internal parameters through a process of minimizing a predefined loss function. The training process involved adjusting the weights of the models' neural network layers, effectively enabling them to learn discriminative features necessary for odometer recognition.

4.1.4 Model Evaluation

To assess the performance of the trained models, the remaining 20% of the dataset was reserved for evaluation. A comprehensive set of metrics, including precision, recall, and mean average precision (mAP), was employed to quantitatively measure the models' accuracy. Moreover, qualitative analysis was performed to visually inspect the models' ability to accurately detect and localize the digits on the odometer. These evaluations provided valuable insights into the models' effectiveness in handling real-world odometer images.

4.1.5 Model Selection

Based on the evaluation results, the model exhibiting the highest accuracy and precision in odometer recognition was selected. Criteria such as high mAP, minimal false positives, and computational efficiency were taken into consideration during the model selection process. The chosen model was deemed the most suitable candidate for subsequent OCR testing, indicating its potential for precise and efficient extraction of numeric information from odometer images.

4.1.6 Optical Character Recognition (OCR) Test

To validate the selected model's capability to accurately extract numeric information, an OCR test was conducted. Real-world odometer images, not previously encountered in the training or evaluation stages, were used to evaluate the model's performance in a more challenging scenario. The model was applied to these unseen images, and the extracted numeric information was compared against the ground truth values. This assessment allowed for the determination of the model's accuracy and identification of potential limitations or challenges.

4.2 Acquisition of Automobile Odometer Data

4.2.1 Using OBD2 adapter

We utilize the OBD2[33] port on market-standard automobiles to retrieve odometer data. The OBD2 port is connected to an OBD2 USB Adapter, which is also connected to the configured raspberry pi. The odometer information is sent to the raspberry pi. The raspberry pi is programmed to receive data from the OBD2 port and transmit it to the onboard computational unit, which will execute the smart contract, at predetermined intervals.

The data received by the Raspberry Pi[34] will be HEX values. With reference to OBD-II PIDs[35], we have programmed the raspberry pi to receive HEX values and convert them to decimal values. The odometer's HEX values are not standard Hexadecimal values. The proper method must be followed when converting and screening the required total miles/kilometers figure. Therefore, the Python package (python-OBD)[36] running on the Raspberry Pi will acquire the value and transmit it via Bluetooth to the onboard computational unit.

4.2.2 Using Odometer Image

Odometer images were obtained directly from the car dashboard using a camera. Multiple images were captured from different angles and distances to ensure variability. The odometer images were seamlessly integrated into the selected deep learning model for data storing. The images served as input data, while the corresponding annotated numeric values were used as ground truth labels. This integration allowed the model to collect the data from the odometer and transfer it to the next step.

4.3 Creation of Smart Contract

The smart contract was developed using the Solidity programming language[37]. Once deployed, our smart contract is capable of completing two operations. Creating a new transaction on the Ethereum Blockchain as well as invoking the transaction to retrieve data from the Blockchain.

Figure 10: Smart Contract Creation Code

4.4 Deploying Smart Contract on Goerli TESTNET

In the previous part we have created the smart contract. Now the smart contract is needed to be deployed on the Ethereum TESTNET. We are using the Goerli TESTNET. The deploy code is written in python[38]. The IDE we are using is Microsoft Visual Code[39].

We have used the Brownie package[40]. The Brownie package significantly reduces the code size and complexity. The Web3.py package[41] can also be used to deploy the smart contract. But it increases the complexity while writing the code. We have also written and deployed the smart contract using the Web3.py package. The following code in Figure 6 is written using Web3.py package. In both Brownie and Web3.py package our smart contract in deploy successfully.

Appendix 3 shows the successful deploy of smart contract. The address at which the contract is deployed is also visible in the figure. The important data set from the above the terminal is summarized in the following table.

Table 2: Smart Contract Information

Smart Contract Address	Priority Fee		Gas Price		Block No
0x3fC26f3D65550F10992A66306ab445Cce6956879	Gwei	USD	Gwei	USD	8226147
	2.0	0.00000309	2.000000349	0.00000309	

4.5 Interaction with the Smart Contract

Our smart contract is implemented on the Blockchain with success (Goerli TESTNET). Smart Contracts provide a variety of blockchain capabilities. After deployment, our smart contract can be accessed via the address listed in the table: 0x3fC26f3D65550F10992A66306b445Cce6956879. The smart contract possesses two capabilities. The first step is to write on the blockchain network, and the second step is to call the written transaction. We will utilize the first one to add the odometer reading and user information to the blockchain.

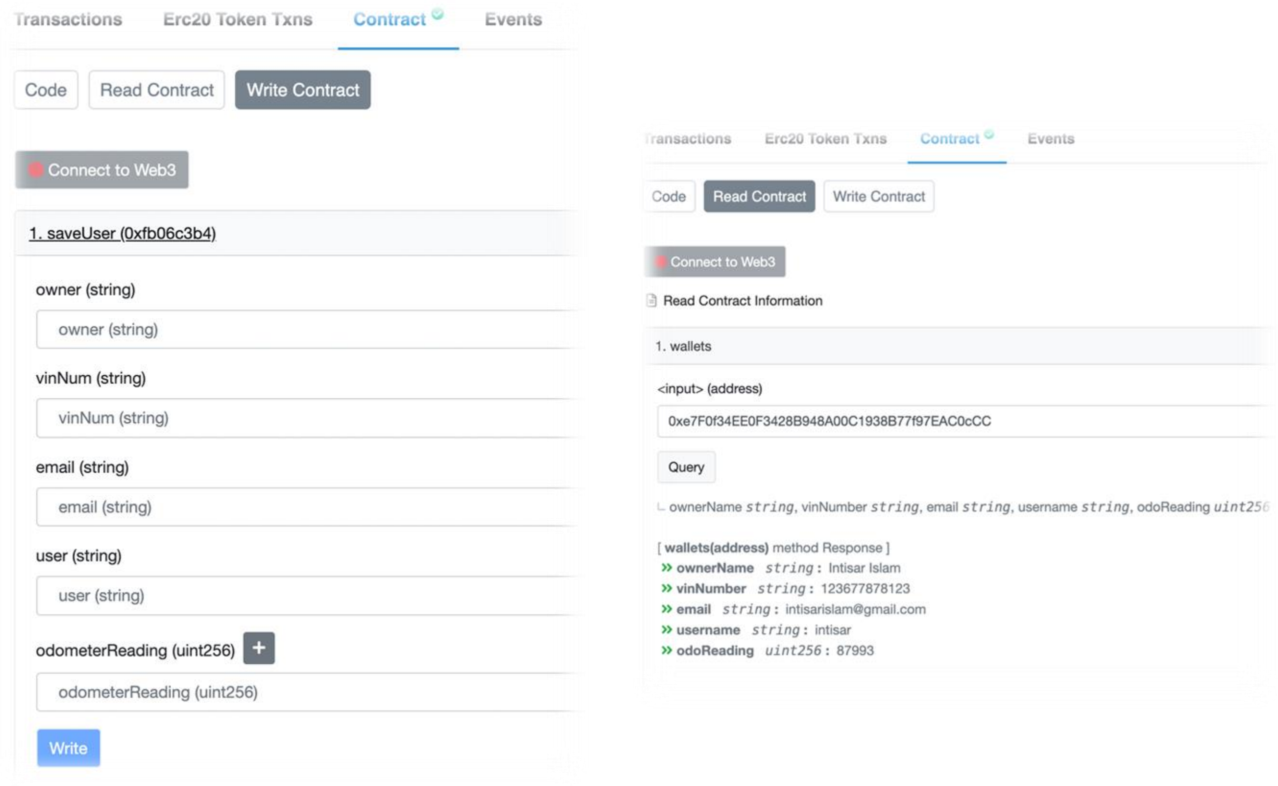


Figure 11: Smart Contract Functionality

Then, using the related PUBLIC address when initiating a transaction on the blockchain, other users can access the data to check the odometer reading.

From the diagram, we can see that the vehicle's owner must connect to Web3 each time he or she wishes to add information to the blockchain. This signifies that the owner must connect to a wallet containing Ethereum cash.

Using the Public Address connected with the wallet, as depicted in the figure 8, a person can access the data entered via this smart contract on the blockchain.

Chapter 5: Results and Discussions

5.1 Faster R-CNN with Region Proposal Networks (RPN)

The training loss graph represents the loss values of the Faster R-CNN model during the training process. The x-axis represents the steps or iterations, while the y-axis represents the training loss values.

Batch size 2

Write here

Training:

At the beginning of training, the loss is relatively high, indicating that the model not yet learned to accurately has the model is improving its object detection and classification abilities.

Around step 5,000, there is a significant drop in the loss, suggesting a breakthrough in the model's learning. This drop indicates that the model has likely learned important features or patterns that contribute to better object detection and classification performance.

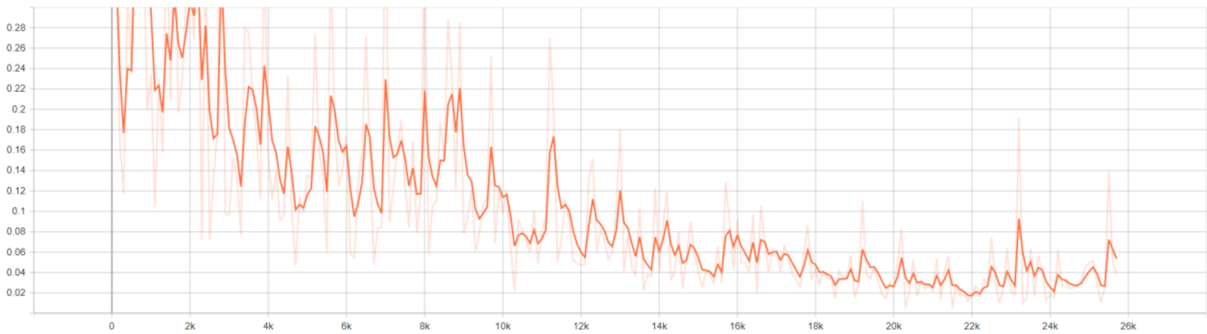


Figure 12 Total Loss

After this initial drop, the loss continues to decrease, but at a slower rate. The gradual decrease indicates that the model is refining its performance and fine-tuning its parameters to better detect and classify objects.

Around step 15,000, there is a noticeable change in the loss curve. The slope of the loss curve becomes less steep, indicating a potential transition from a more rapid improvement phase to a slower convergence phase. This change suggests that the model has captured most of the important features related to object detection and classification and is now focusing on fine-tuning and converging to a more optimal solution.

Here the training has been conducted for 25k steps and the batch size is 2. As it can been seen from the graph the training loss reached 0.04 but also there is a lot of noise in the training curve. The lower batch size is a probable cause.

Now, the model has been trained for about 50,000 steps and the training loss reaches as low as 0.02 . Noise in the traning curve has also decreased due to the longer training time and steps.

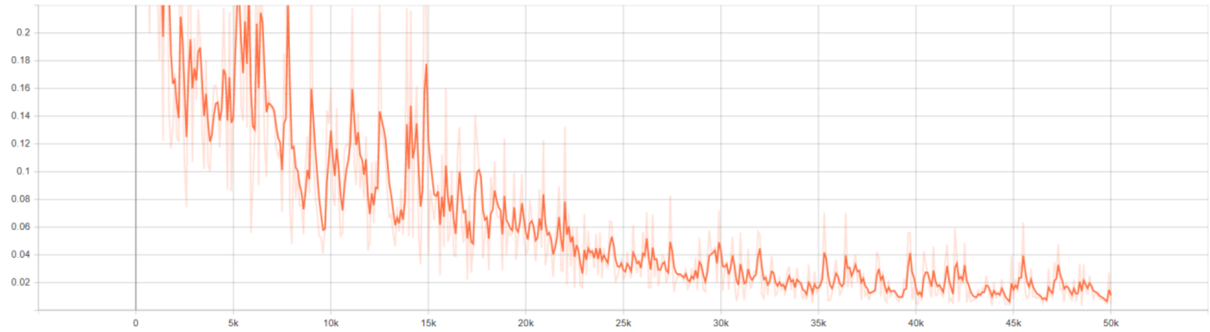


Figure 13 Total Loss

Batch size 4

Training:

In this training session, we have increased the batch size to 4 and the significance reduction of noise in the training curve can be noticed. Here the model has been trained for only 25k steps and it reduces the training loss to 0.02 (2%) which is half of the steps of previous

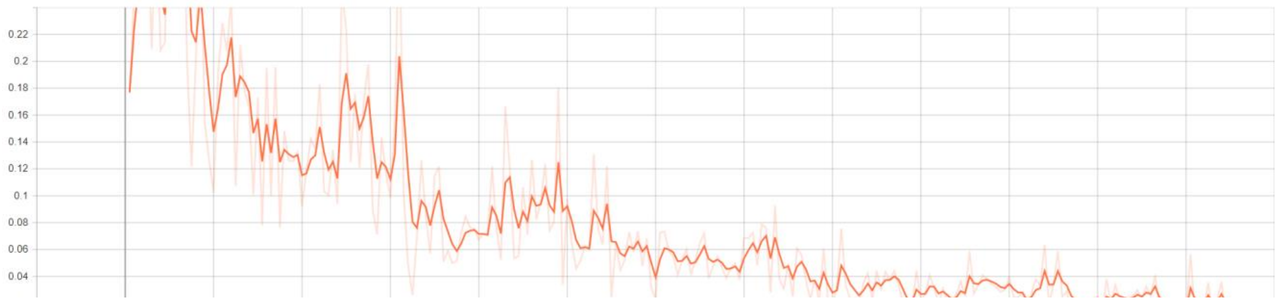


Figure 14 Total Loss

training session with batch size of 2.

The training loss graph represents the loss values of the Faster R-CNN model trained with a batch size of 4. The x-axis represents the steps or iterations during the training process, while the y-axis represents the training loss values.

In this graph, we can observe a similar pattern to the previous descriptions. At the beginning of training, the loss is relatively high, indicating that the model hasn't yet learned to accurately detect and classify objects.

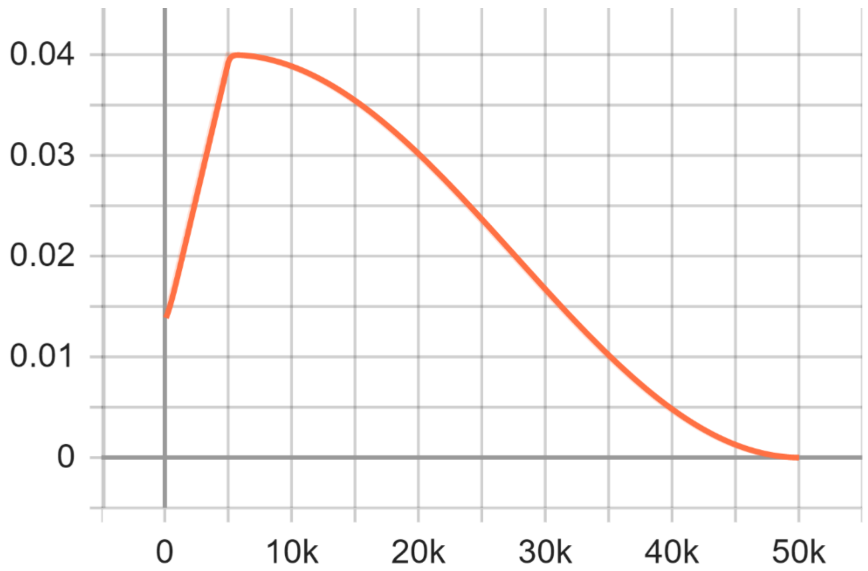
Around step 5,000, there is a significant drop in the loss, suggesting that the model has made progress in learning important features or patterns related to object detection and classification.

As the training progresses, the loss continues to decrease, indicating that the model is refining its performance and improving its ability to detect and classify objects.

Around step 15,000, there may be a noticeable change in the loss curve. The slope of the loss curve becomes less steep, indicating a potential transition from a more rapid improvement phase to a slower convergence phase. This change suggests that the model has captured most of the important features and is focusing on fine-tuning and converging to a more optimal solution.

Learning rate:

This training session learning rate is Cosine Decay learning rate. During the initial phase the training learning rate warms up and quickly increases its learning rate within 5,000 to a maximum of 0.04 then decreases slowly until the end of the training.



Batch size 6

Due to the promising results of increasing batch size we decided to increase the batch size to 6. This model was trained on Nvidia Tesla T4 GPU.

Training:

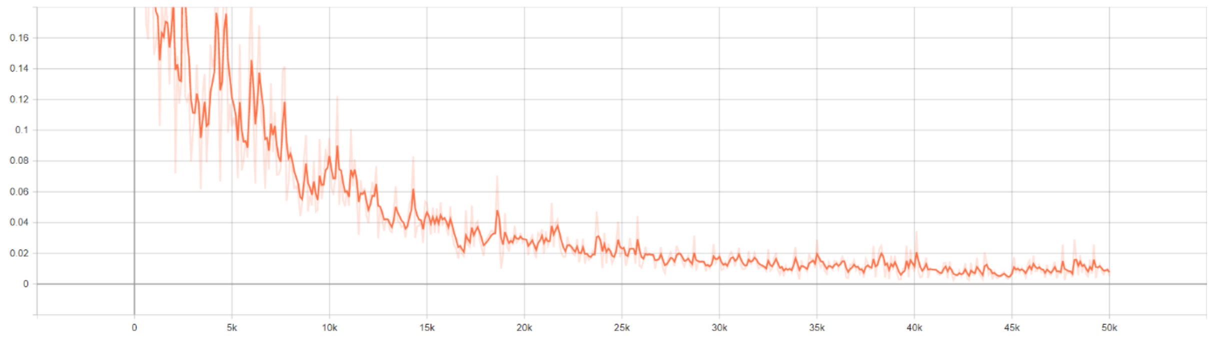


Figure 15 Total Loss

From the graph, we can observe similar trends to the previous descriptions. At the beginning of training, the loss is relatively high, indicating that the model is in the early stages of learning and hasn't yet achieved accurate object detection and classification.

Around step 5,000, there might be a significant drop in the loss, suggesting that the model has made progress in learning crucial features or patterns for object detection and classification.

As training progresses, the loss continues to decrease, indicating that the model is refining its performance and enhancing its ability to detect and classify objects.

Around step 15,000, there may be a noticeable change in the loss curve. The slope of the loss curve becomes less steep, indicating a potential transition from a more rapid improvement phase to a slower convergence phase. This suggests that the model has captured most of the important features and is focusing on fine-tuning and converging towards a more optimal solution.

While the overall patterns and trends in the loss curve are likely to be similar to the previous graphs, the specific values and convergence points may differ due to variations in batch size and other factors. Without the previous graphs for direct comparison, it's challenging to provide a detailed analysis of the differences and similarities between the training processes.

The training loss reaches as low as 0.005 (0.5%) and the noise in the training curve has significantly dropped. A point to notice is that the low amount of noise in training curve indicates that the possibility of reaching the global minimum. Batch size 6 has been trained for 50,000 steps.

Learning rate:

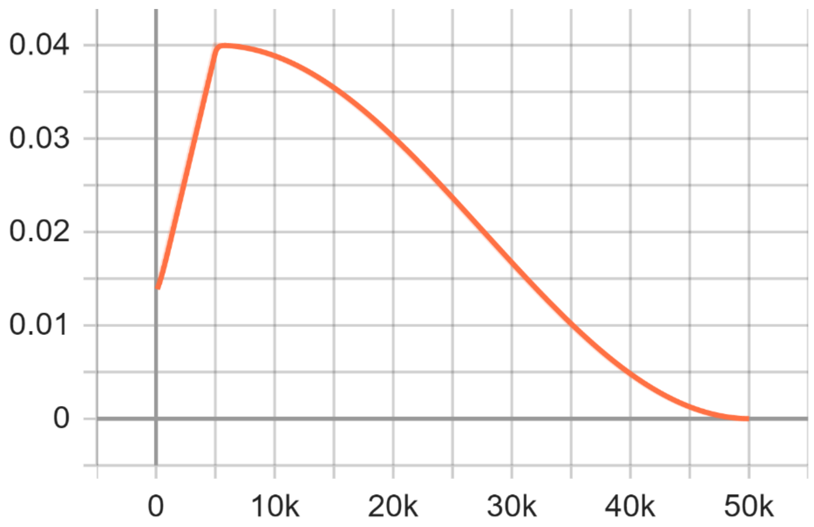


Figure 16 Learning Rate

The learning rate follows the same pattern as the previous training learning rate. This is also Cosine Decay Learning Rate. Warmup steps 5,000.

5.2 CenterNet: Keypoint Triplets for Object Detection

Write here

Batch size 6

Write here

Training:

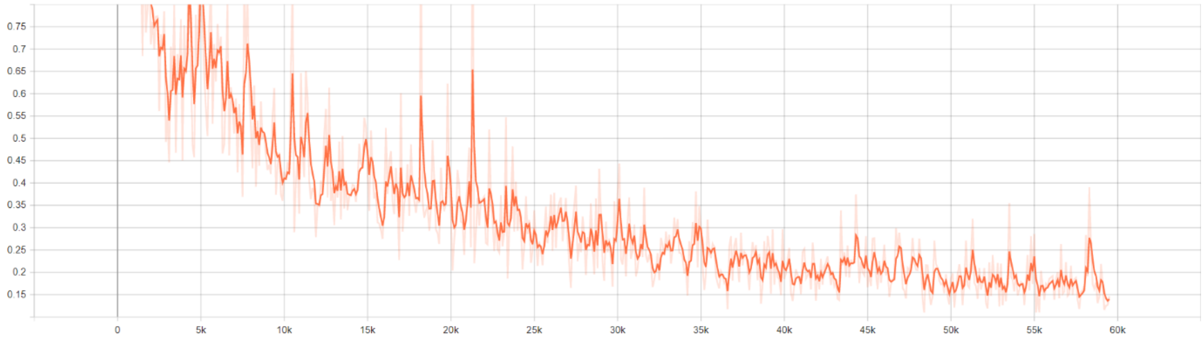


Figure 17 Total Loss

The graph represents the training loss over 60,000 steps, with a batch size of 6. The x-axis represents the steps or iterations during the training process, starting from 0 and going up to 60,000. The y-axis represents the corresponding training loss values.

At the beginning of the training, around step 0, the loss value is relatively high. As the training progresses, the loss steadily decreases, indicating that the model is learning and improving its performance. The decreasing trend continues until approximately step 15,000, where the loss appears to stabilize at a lower value.

After step 15,000, there are some fluctuations in the loss values. These fluctuations suggest that the model may be fine-tuning or converging to a local minimum in the loss landscape. Despite these fluctuations, the overall trend of decreasing loss is maintained.

Around step 55,000, the loss starts to plateau, indicating that the model's performance is not significantly improving with further training. This plateauing effect suggests that the model has reached a point where additional training may not yield substantial improvements.

Learning rate:

From the beginning of training until approximately step 10,000, the learning rate remains constant at a relatively high value. This higher learning rate is typically used to facilitate the initial phase of training, allowing the model to make larger adjustments to its parameters.

Around step 10,000, there is a sudden drop in the learning rate. This drop indicates a learning rate schedule or a specific strategy implemented during training. Lowering the learning rate at this stage is often done to enable the model to fine-tune and converge to a more optimal

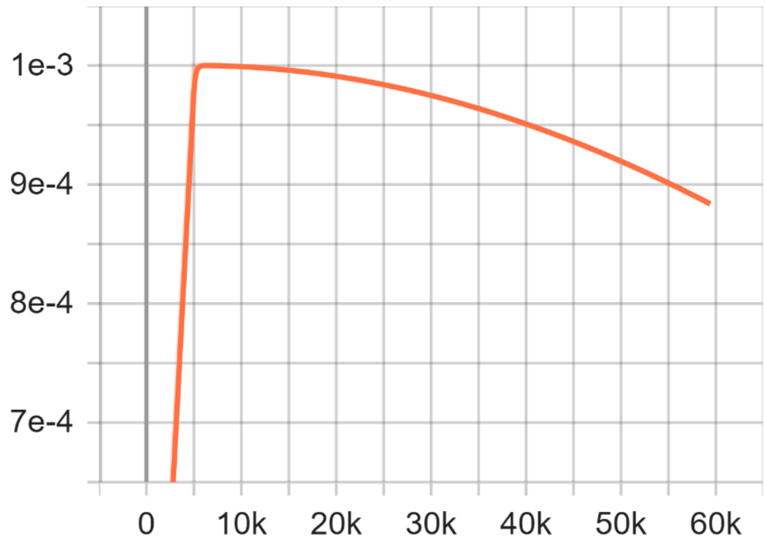


Figure 18 Learning Curve

solution.

After the initial drop, the learning rate continues to decrease gradually. This gradual decrease is commonly applied to help the model converge slowly and refine its performance over time. The diminishing learning rate allows for smaller parameter updates as the model approaches convergence.

The learning rate graph provides insights into the learning rate schedule used during training. It shows how the learning rate is adjusted over the course of training to balance between rapid exploration of the parameter space and fine-tuning to converge to a good solution.

5.3 SSD: Single Shot MultiBox Detector

The training loss curve represents the loss values of an SSD (Single Shot MultiBox Detector) object detection model trained for 40,000 steps, with a batch size of 6. The x-axis represents the steps or iterations during the training process, while the y-axis represents the training loss values.

Batch size 6

Training:

At the beginning of training, the loss is relatively high, indicating that the model has not yet learned to accurately detect objects. As the training progresses, the loss gradually decreases, indicating that the model is improving its object detection capabilities.

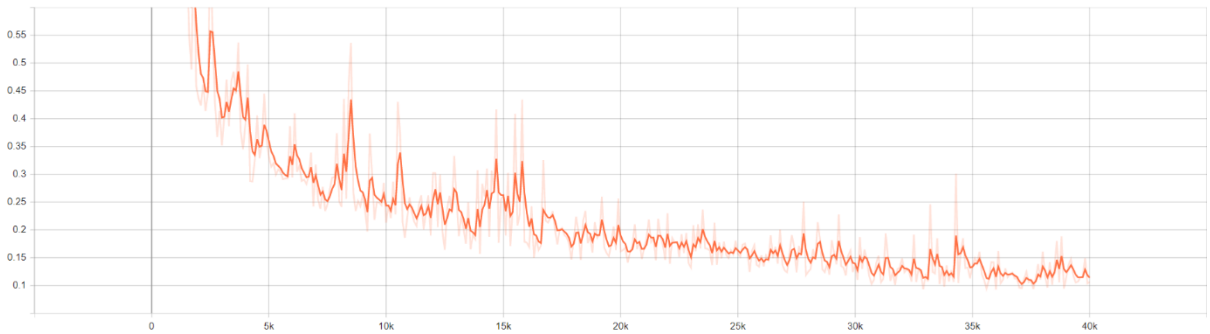


Figure 19 Total Loss

Around step 5,000, there is a significant drop in the loss, suggesting a breakthrough in the model's learning. This drop indicates that the model has likely learned important features or patterns that contribute to better object detection accuracy.

After this initial drop, the loss continues to decrease but at a slower rate. The gradual decrease indicates that the model is steadily refining its object detection performance, fine-tuning its parameters, and improving its ability to locate and classify objects in the input data.

Around step 15,000, there is another significant change in the loss curve. The slope of the loss curve becomes less steep, indicating a potential transition from a more rapid improvement phase to a slower convergence phase. This change suggests that the model may have captured most of the important features related to object detection and is now focusing on fine-tuning and converging to a more optimal solution.

As the training progresses further, the loss continues to decrease, but the rate of improvement diminishes. This decrease may indicate that the model is reaching a point where it is more challenging to further enhance the object detection performance. The loss curve starts to settle and reach a plateau.

Around step 30,000, the loss curve appears to flatten, suggesting that the model's performance has converged to a stable point. At this stage, additional training may not lead to significant improvements in the object detection accuracy.

Learning rate:

At the beginning of training, the learning rate is relatively high, which allows the model to

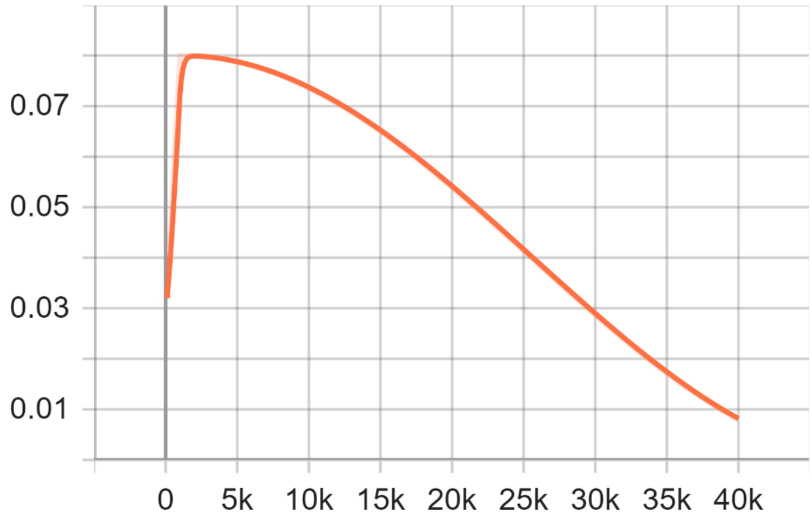


Figure 20 Learning Curve

make larger updates to its parameters and explore the parameter space more extensively. This higher learning rate helps the model quickly learn initial representations and patterns.

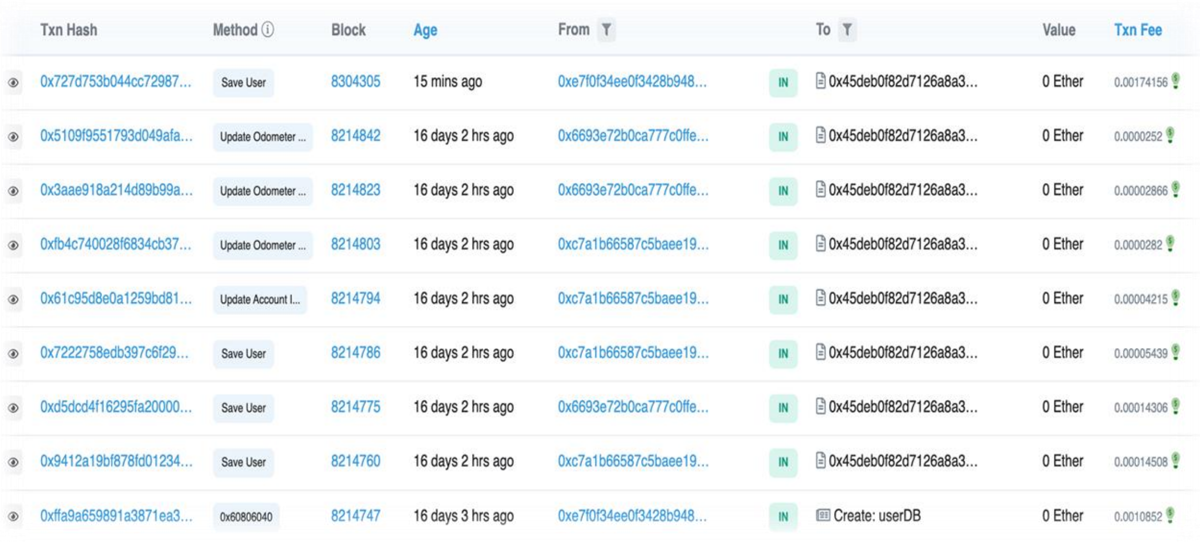
Around step 5,000, there is a significant drop in the learning rate. This drop indicates a change in the learning rate schedule or a specific strategy employed during training. Lowering the learning rate at this point is often done to facilitate fine-tuning and help the model converge to a more optimal solution. The drop in the learning rate signifies a transition from the initial phase of exploration to a phase focused on fine-tuning and convergence. Following the initial drop, the learning rate continues to decrease gradually over the course of training. This gradual decrease is typically employed to refine the model's performance and allow for smaller updates to the model's parameters. The diminishing learning rate helps the model converge slowly and improve its object detection capabilities.
Around step 30,000, there is a noticeable decrease in the learning rate. This reduction suggests that the model has likely reached a stage where it requires smaller adjustments to its parameters to fine-tune and optimize its performance. The lower learning rate allows the model to make more precise updates, ensuring it converges towards a good solution.

Table 3 MAP (Mean Average Precision) comparison

Model Name	Training steps	Batch size	MAP	Evaluation metrics
Faster R-CNN with Region Proposal Networks (RPN)	25000	2	.657688 (65.76%)	
	25000	4	.674588 (67.75%)	
	50000	2	.683400 (68.33%)	
	50000	6	.702900 (70.29%)	
SSD: Single Shot MultiBox Detector	40000	8	.447867 (44.78%)	
CenterNet: Keypoint Triplets for Object Detection	60000	6	.626557 (62.65%)	

5.4 Block Creation time & gas fees associated with the vehicle data entry

Four vehicle records have been registered via our smart contract onto the blockchain. Each entry of data is regarded as a transaction. Every transaction has a corresponding transaction time and charge. The following diagram displays information on the establishment of smart contracts, the entry and update of car odometer data, and the update of user information.



Txn Hash	Method	Block	Age	From		To	Value	Txn Fee
0x727d753b044cc72987...	Save User	8304305	15 mins ago	0xe7f0f34ee0f3428b948...	IN	0x45deb0f82d7126a8a3...	0 Ether	0.00174156
0x5109f9551793d049afa...	Update Odometer ...	8214842	16 days 2 hrs ago	0x6693e72b0ca777c0ffe...	IN	0x45deb0f82d7126a8a3...	0 Ether	0.0000252
0x3aae918a214d89b99a...	Update Odometer ...	8214823	16 days 2 hrs ago	0x6693e72b0ca777c0ffe...	IN	0x45deb0f82d7126a8a3...	0 Ether	0.00002866
0xfb4c740028f6834cb37...	Update Odometer ...	8214803	16 days 2 hrs ago	0xc7a1b66587c5baee19...	IN	0x45deb0f82d7126a8a3...	0 Ether	0.0000282
0x61c95d8e0a1259bd81...	Update Account I...	8214794	16 days 2 hrs ago	0xc7a1b66587c5baee19...	IN	0x45deb0f82d7126a8a3...	0 Ether	0.00004215
0x7222758edb397c6f29...	Save User	8214786	16 days 2 hrs ago	0xc7a1b66587c5baee19...	IN	0x45deb0f82d7126a8a3...	0 Ether	0.00005439
0xd5dcd4f16295fa20000...	Save User	8214775	16 days 2 hrs ago	0x6693e72b0ca777c0ffe...	IN	0x45deb0f82d7126a8a3...	0 Ether	0.00014306
0x9412a19bf878fd01234...	Save User	8214760	16 days 2 hrs ago	0xc7a1b66587c5baee19...	IN	0x45deb0f82d7126a8a3...	0 Ether	0.00014508
0xffa9a659891a3871ea3...	0x60806040	8214747	16 days 3 hrs ago	0xe7f0f34ee0f3428b948...	IN	Create: userDB	0 Ether	0.0010852

Figure 21: Activity on the Blockchain

If we summarize the figure 9, we can present the table below for easier comprehension. The table summarizes the relevant actions, transaction duration, and transaction fees. No transaction is required to retrieve odometer data. The query requires only the public address from which the odometer information transaction occurred. We have developed our deploy code to give a minimal Priority fee for odometer data to be included in the blockchain at a significantly faster rate. Although it raises the cost to include odometer data on the blockchain, it drastically decreases the time required to incorporate transaction verification on the blockchain. From the Table 3 & Figure 9, we have conclusive evidence that our smart contract is able to continually add odometer data to the blockchain and conduct blockchain queries as needed without incurring transaction fees

Table 4: Summary of the activities being conducted through our smart contract on the blockchain

Action	**Block Number**	**Time (t)**	**Age**	**Transaction Fee (Ether)**
Smart Contract Deploy	8214747	120s	16 days 20 hrs	0.0010852
Odometer data entry	8214760	35s	16 days 20 hrs	0.00014508
Odometer data entry	8214775	45s	16 days 20 hrs	0.00014306
Odometer data entry	8214786	46s	16 days 20 hrs	0.00005439
Odometer data update	8214803	34s	16 days 20 hrs	0.0000282
Odometer data update	8214823	74s	16 days 20 hrs	0.00002866
Odometer data update	8214842	43s	16 days 20 hrs	0.0000252

Chapter 6 Conclusion

6.1 Conclusion

We have successfully constructed and deployed the smart contract on the blockchain. Physical completion of the vehicle odometer data acquisition system is still pending. Consequently, we can assert that the Smart Contract creation & deploy of our system has been successfully executed and its functionality has been verified. We must construct the initial portion of our system, in which odometer data will be automatically relayed to the onboard computing unit and sent to the blockchain. Additionally, it is necessary to guarantee that the raspberry pi that will be installed on the car is capable of performing specific duties to enhance security.

6.2 Future Scopes

The future scope of the project holds great potential in several key areas. Industry collaboration will play a crucial role in advancing the project, as partnering with other organizations and stakeholders will allow for the exchange of knowledge, resources, and expertise. This collaboration will foster innovation, drive efficiency, and unlock new opportunities for growth. Additionally, extended fraud detection capabilities will be a key focus, leveraging advanced technologies such as machine learning and artificial intelligence to proactively identify and prevent fraudulent activities. This will enhance the project's integrity and build trust among users. Furthermore, data analytics and predictive models will enable the project to harness the power of big data, gaining valuable insights, and making informed decisions. These analytical capabilities will facilitate better risk management, operational efficiency, and strategic planning. Lastly, building our own blockchain network will offer enhanced security, transparency, and decentralization, revolutionizing how transactions and data are stored and verified. This blockchain network will provide a solid foundation for the project's long-term scalability and sustainability, empowering users with greater control and privacy. Overall, these future prospects demonstrate the project's commitment to continuous innovation and its potential to transform industries and drive positive change.

References

[1] C. A. Leke and T. Marwala, "Introduction to deep learning," *Studies in Big Data*, vol. 48, pp. 21–40, Feb. 2020, doi: 10.1007/978-3-030-01180-2_2.

[2] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, "Untangling Blockchain: A Data Processing View of Blockchain Systems," *IEEE Trans Knowl Data Eng*, vol. 30, no. 7, pp. 1366–1385, Jul. 2018, doi: 10.1109/TKDE.2017.2781227.

[3] L. R. Abbade *et al.*, "Blockchain Applied to Vehicular Odometers," *IEEE Netw*, vol. 34, no. 1, pp. 62–68, Jan. 2020, doi: 10.1109/MNET.001.1900162.

[4] R. V. Enrico Pastori, "Research for TRAN Committee - Odometer tampering: measures to prevent it," *Policy Commons*, 2017. https://policycommons.net/artifacts/2055364/research-for-tran-committee-odometer-tampering/2808455/

[5] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," *Int J Res Appl Sci Eng Technol*, vol. 10, no. 12, pp. 943–947, Nov. 2015, doi: 10.22214/ijraset.2022.47789.

[6] F. Zhuang *et al.*, "A Comprehensive Survey on Transfer Learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, Nov. 2019, doi: 10.1109/JPROC.2020.3004555.

[7] D. Vujičić, D. Jagodić, and S. Randić, "Blockchain technology, bitcoin, and Ethereum: A brief overview," *2018 17th International Symposium on INFOTEH-JAHORINA, INFOTEH 2018 - Proceedings*, vol. 2018-January, pp. 1–6, Apr. 2018, doi: 10.1109/INFOTEH.2018.8345547.

[8] "Ethereum (ETH) Blockchain Explorer." https://etherscan.io/ (accessed Jan. 15, 2023).

[9] B. Version, "ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER".

[10] "Home · ethereum/wiki Wiki." https://github.com/ethereum/wiki/wiki/ (accessed Jan. 15, 2023).

[11] "ERC20 Token Standard - IndexUniverse Crypto." https://www.indexuniverse.eu/erc20-token-standard/ (accessed Jan. 15, 2023).

[12] "Groupe Renault teams with Microsoft and Viseo to... - Google Scholar." https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Groupe+Renault+teams+with+Microsoft+and+Viseo+to+create+the+first-ever+digital+car+maintenance+book+prototype&btnG= (accessed Jan. 15, 2023).

[13] "Decentralized Vehicle History by VIN | VINchain.io." https://vinchain.io/ (accessed Jan. 15, 2023).

[14] "carVertical | VIN decoder, check a car and get vehicle history report." https://www.carvertical.com/en (accessed Jan. 15, 2023).

[15] M. Chanson, E. Fleisch, A. Bogner, and F. Wortmann, "Blockchain as a privacy enabler: An odometer fraud prevention system," *UbiComp/ISWC 2017 - Adjunct Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, pp. 13–16, Sep. 2017, doi: 10.1145/3123024.3123078.

[16] M. Vasile and B. Groza, "DeMetrA - Decentralized metering with user anonymity and layered privacy on blockchain," *2019 23rd International Conference on System Theory, Control and Computing, ICSTCC 2019 - Proceedings*, pp. 560–565, Oct. 2019, doi: 10.1109/ICSTCC.2019.8885761.

[17] K. Leo Brousmiche, A. Durand, T. Heno, C. Poulain, A. Dalmieres, and E. ben Hamida, "Hybrid Cryptographic Protocol for Secure Vehicle Data Sharing Over a Consortium Blockchain," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, IEEE, Jul. 2018, pp. 1281–1286. doi: 10.1109/Cybermatics_2018.2018.00223.

[18] G. Saldamli, K. Karunakaran, V. K. Vijaykumar, W. Pan, S. Puttarevaiah, and L. Ertaul, "Securing Car Data and Analytics using Blockchain," *2020 7th International Conference on Software Defined Systems, SDS 2020*, pp. 153–159, Apr. 2020, doi: 10.1109/SDS49854.2020.9143914.

[19] Maciej Serda *et al.*, "Synteza i aktywność biologiczna nowych analogów

tiosemikarbazonowych chelatorów żelaza,” *Uniwersytet śląski*, vol. 7, no. 1, pp. 343–354, 2013, doi: 10.2/JQUERY.MIN.JS.
[20] “‘Trading Real-World Assets on Blockchain - An Application of Trust-Free’ by Benedikt Notheisen 60848387, Jacob Benjamin Cholewa et al.” https://aisel.aisnet.org/bise/vol59/iss6/4/ (accessed Jan. 15, 2023).
[21] S. Smys and H. Wang, “Security Enhancement in Smart Vehicle Using Blockchain-based Architectural Framework,” *Journal of Artificial Intelligence and Capsule Networks*, 2021, doi: 10.36548/jaicn.2021.2.002.
[22] K. L. Brousmiche, T. Heno, C. Poulain, A. Dalmieres, and B. Hamida, “Digitizing, Securing and Sharing Vehicles Life-cycle Over a Consortium Blockchain: Lessons Learned”, Accessed: Apr. 26, 2023. [Online]. Available: https://hal.science/hal-01760781
[23] M. Chanson, A. Bogner, D. Bilgeri, E. Fleisch, F. Wortmann, and E. Zurich, “Privacy-Preserving Data Certification in the Internet of Things: Leveraging Blockchain Technology to Protect Sensor Data”.
[24] “IEEE Xplore Full-Text PDF:” https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8977438 (accessed Apr. 26, 2023).
[25] M. Qatawneh and S. El-Switi, “Application of Blockchain Technology in Used Vehicle Market: A Review Parallel Matrix Multiplication Algorithm on Hex-Cell network View project Application of Blockchain Technology in Used Vehicle Market: A Review,” 2021, doi: 10.1109/ICIT52682.2021.9491670.
[26] M. Gösele and P. Sandner, “Analysis of blockchain technology in the mobility sector,” *Forschung im Ingenieurwesen/Engineering Research*, vol. 83, no. 4, pp. 809–816, Dec. 2019, doi: 10.1007/S10010-019-00315-Y/TABLES/5.
[27] G. Falco, J. Siegel, and J. E. Siegel, “A Distributed`BlackDistributed`Black Box’ Audit Trail Design Specification for Connected and Automated Vehicle Data and Software Assurance NeuroMesh IoT Security View project Cloud vehicle mirroring View project ASSURING AUTOMOTIVE DATA AND SOFTWARE INTEGRITY EMPLOYING DISTRIBUTED HASH TABLES AND BLOCKCHAIN A PREPRINT,” 2020, Accessed: Apr. 26, 2023. [Online]. Available: https://www.researchgate.net/publication/339139299
[28] “The PASCAL Visual Object Classes Homepage.” http://host.robots.ox.ac.uk/pascal/VOC/ (accessed May 19, 2023).
[29] C. L. R. G. X. Y. W. L. J. Z. Y. B. Z. Y. Y. Y. Q. D. H. W. Yuning Du, “PP-OCR: A Practical Ultra Lightweight OCR System”.
[30] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, “CenterNet: Keypoint Triplets for Object Detection,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-October, pp. 6568–6577, Apr. 2019, doi: 10.1109/ICCV.2019.00667.
[31] W. Liu *et al.*, “SSD: Single Shot MultiBox Detector,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9905 LNCS, pp. 21–37, Dec. 2015, doi: 10.1007/978-3-319-46448-0_2.
[32] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans Pattern Anal Mach Intell*, vol. 39, no. 6, pp. 1137–1149, Jun. 2015, doi: 10.1109/TPAMI.2016.2577031.
[33] “On-board diagnostics - Wikipedia.” https://en.wikipedia.org/wiki/On-board_diagnostics (accessed Jan. 15, 2023).
[34] “Raspberry Pi Documentation.” https://www.raspberrypi.com/documentation/ (accessed Jan. 15, 2023).
[35] “OBD-II PIDs - Wikipedia.” https://en.wikipedia.org/wiki/OBD-II_PIDs (accessed Jan. 15, 2023).
[36] “Getting Started - python-OBD.” https://python-obd.readthedocs.io/en/latest/ (accessed Jan. 15, 2023).
[37] “Solidity — Solidity 0.8.17 documentation.” https://docs.soliditylang.org/en/v0.8.17/ (accessed Jan. 15, 2023).
[38] “3.11.1 Documentation.” https://docs.python.org/3/ (accessed Jan. 15, 2023).
[39] “Documentation for Visual Studio Code.” https://code.visualstudio.com/docs (accessed Jan. 15, 2023).

[40] "Brownie — Brownie 1.19.2 documentation." https://eth-brownie.readthedocs.io/en/stable/ (accessed Jan. 15, 2023).
[41] "Introduction — Web3.py 5.31.3 documentation." https://web3py.readthedocs.io/en/v5/ (accessed Jan. 15, 2023).

Appendix

01 Smart Contract Creation Code

```
// SPDX-License_Identifier: MIT

pragma solidity >=0.6.0 <0.9.0;

contract userDB {
    struct userinfo {
        string ownerName;
        string vinNumber;
        string email;
        string username;
        uint256 odoReading;
    }
    mapping(address => userinfo) public wallets;

    function saveUser(
        string calldata owner,
        string calldata vinNum,
        string calldata email,
        string calldata user,
        uint256 odometerReading
    ) external {
        address wallet = msg.sender;
        wallets[wallet].ownerName = owner;
        wallets[wallet].vinNumber = vinNum;
        wallets[wallet].email = email;
        wallets[wallet].username = user;
        wallets[wallet].odoReading = odometerReading;
    }

    function updateOdometerReading(uint256 data) external {
        address wallet = msg.sender;
        if (data < wallets[wallet].odoReading) {
            wallets[wallet].odoReading = wallets[wallet].odoReading;
        } else {
            wallets[wallet].odoReading = data;
        }
    }

    function updateAccountInfo(
        string calldata owner,
        string calldata email,
        string calldata user
    ) external {
        address wallet = msg.sender;
        wallets[wallet].ownerName = owner;
        wallets[wallet].email = email;
```

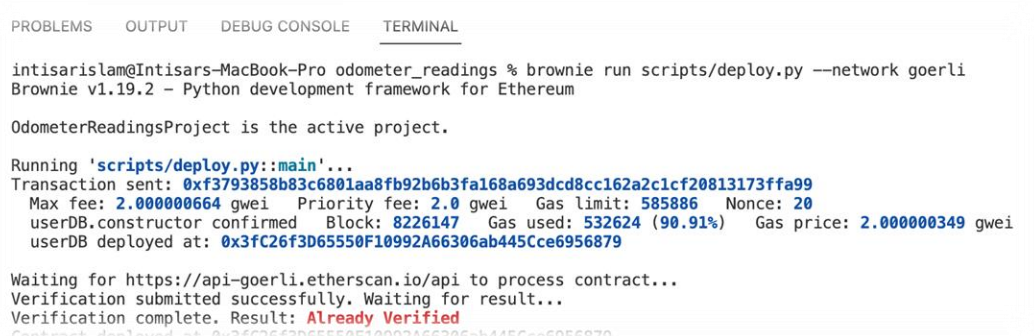
02 Deploy Code using Brownie Package

```python
from brownie import userDB
from brownie import network, config, accounts
from brownie.network import priority_fee

priority_fee("2 gwei")


def deplpy_odometer():
    account = get_account()
    odometer_info = userDB.deploy({"from": account}, publish_source=True)
    print(f"Contract deployed at {odometer_info.address}")


def get_account():
    if network.show_active() == "development":
        return accounts[0]
    else:
        return accounts.add(config["wallets"]["from_key"])

def main():
    deplpy_odometer()
```

03 Deploy Code using Web3.py

```python
from solcx import compile_standard, install_solc
import json
from web3 import Web3
import os
from dotenv import load_dotenv

load_dotenv()

with open("./SimpleStorage.sol", "r") as file:
    simple_storage_file = file.read()

# We add these two lines that we forgot from the video!
print("Installing...")
install_solc("0.6.0")
# Solidity source code
compiled_sol = compile_standard(
    {
        "language": "Solidity",
        "sources": {"SimpleStorage.sol": {"content": simple_storage_file}},
        "settings": {
            "outputSelection": {
                "*": {
                    "*": ["abi", "metadata", "evm.bytecode", "evm.bytecode.sourceMap"]
                }
            }
        },
    },
    solc_version="0.6.0",
)

with open("compiled code.json", "w") as file:
    json.dump(compiled_sol, file)

# get bytecode
bytecode = compiled_sol["contracts"]["SimpleStorage.sol"]["SimpleStorage"]["evm"][
    "bytecode"
]["object"]

# get abi
abi = compiled_sol["contracts"]["SimpleStorage.sol"]["SimpleStorage"]["abi"]

# for connecting to ganache
w3 = Web3(Web3.HTTPProvider("http://127.0.0.1:8545"))
chain_id = 1337
my_address = "0x0334566cb1e1dCB76ac8ed3FAef665d6782B75e8"
private_key = os.getenv("PRIVATE_KEY")
print(private_key)
print(os.getenv("SOME"))
```

04 Deploy Terminal



```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

intisarislam@Intisars-MacBook-Pro odometer_readings % brownie run scripts/deploy.py --network goerli
Brownie v1.19.2 - Python development framework for Ethereum

OdometerReadingsProject is the active project.

Running 'scripts/deploy.py::main'...
Transaction sent: 0xf3793858b83c6801aa8fb92b6b3fa168a693dcd8cc162a2c1cf20813173ffa99
  Max fee: 2.000000664 gwei   Priority fee: 2.0 gwei   Gas limit: 585886   Nonce: 20
  userDB.constructor confirmed   Block: 8226147   Gas used: 532624 (90.91%)   Gas price: 2.000000349 gwei
  userDB deployed at: 0x3fC26f3D65550F10992A66306ab445Cce6956879

Waiting for https://api-goerli.etherscan.io/api to process contract...
Verification submitted successfully. Waiting for result...
Verification complete. Result: Already Verified
```