DETECTION OF TRAFFIC RULES VIOLATION USING BLOCKCHAIN

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

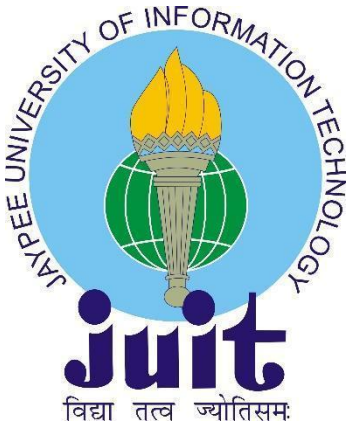
By

Anushka Gupta (191418)
Lalita Gupta (191327)

Under the supervision of

Dr. Amit Kumar

to



Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

CERTIFICATE

We hereby declare that the work presented in this report entitled " Detection of traffic rules violation using blockchain" in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering/Information Technology submitted in the department of Computer Science and Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of our own work carried out over a period from July 2022 to May 2023 under the supervision of Dr. Amit Kumar, Assistant Professor (SG), Department of CSE. The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature)

Student Name:

Rollno.:

(Student Signature)

Student Name:

Rollno.:

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

(Supervisor Signature)

Supervisor Name: Dr. Amit Kumar

Designation: Assistant Professor (SG)

Department name: Department of CSE

Dated:

PLAGIARISM REPORT

ACKNOWLEDGEMENT

Firstly, we express our heartiest thanks and gratefulness to almighty God for his divine blessing makes it possible for us to complete the project work successfully. We are really grateful and wish my profound indebtedness to Supervisor Dr. Amit Kumar, Assistant Professor (SG), Department of CSE Jaypee University of Information Technology, Wakhnaghat. Deep Knowledge and keen interest of my supervisor in the field of "Image Detection and Blockchain" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to Dr. Amit Kumar, Department of CSE, for his kind help to finish my project. I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, we must acknowledge with due respect the constant support and patients of my parents.

Anushka Gupta
Lalita Gupta

TABLE OF CONTENT

LIST OF FIGURES

LIST OF GRAPHS

LIST OF TABLES/ FLOWCHART

ABSTRACT

India is a country with the second largest road network in the world. The Indian government already faces an extremely challenging task in constructing world-class highways because of the sheer magnitude of the project. Indian highways are constantly busy with traffic, and it is usual to see people breaking the rules of the road. It has now placed the burdensome responsibility of keeping an eye out for such violators and punishing them on the shoulders of the traffic authority. High volumes of traffic can result from the growing number of cars in cities, which suggests that traffic offenses are becoming an increasingly serious problem worldwide nowadays. Systems for detecting traffic violations and storing the details are required to address the worrying issue and stop such unthinkable consequences, traffic rule violation detection is needed. In order to prevent this, the system consistently enforcing relevant traffic laws and detaining individuals who disregard them. A real-time traffic infraction detection system is necessary since law enforcement officers are always keeping an eye on the roads. Because the traffic detecting system detects violations faster than people, traffic enforcement personnel will have no trouble enforcing safe roads precisely and effectively.

So, I have made this project that will assist us in identifying the vehicle defaulter who disobeys traffic regulations. Also, the vehicle number plate of the defaulter who disobeys traffic laws is stored into on the blockchain and is accessible in the future.

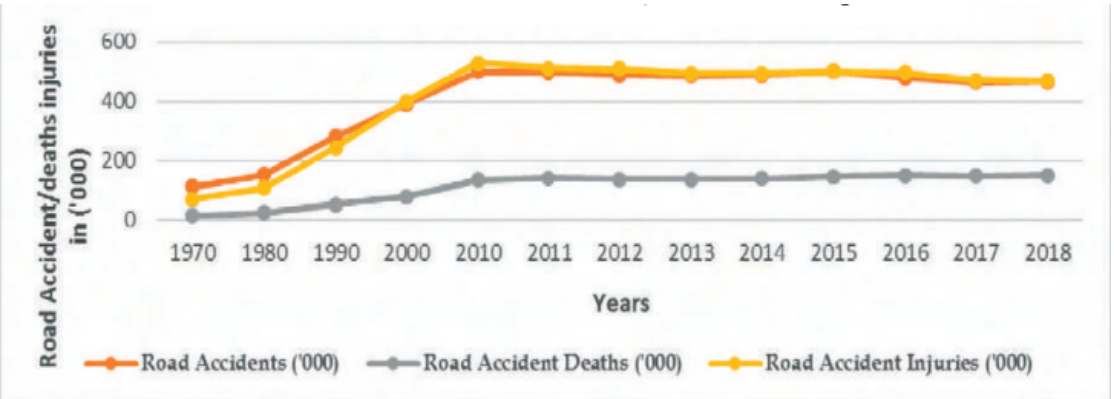
CHAPTER-1: INTRODUCTION

1.1 Introduction

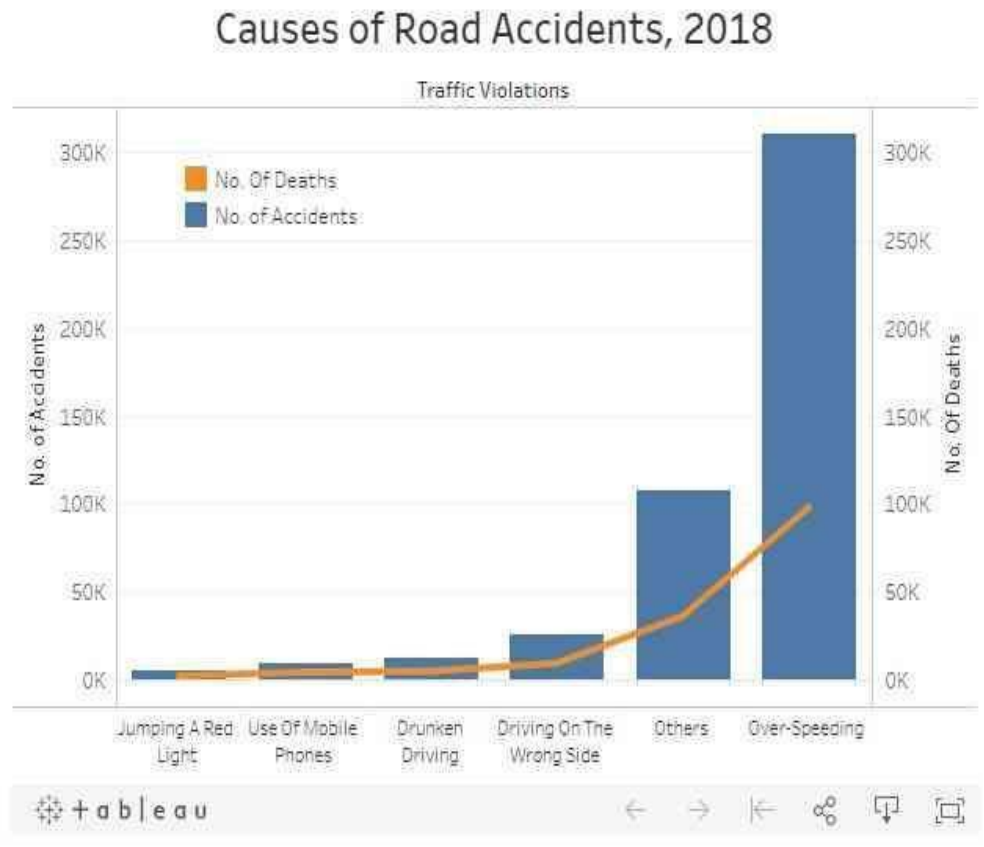
The purpose of traffic laws is to protect commuters from harm on the roadways. But tragically, in India, the majority of people either don't know a lot of the regulations or have a lot of misconceptions about what it means to follow the rules. That is also one of the main causes of India's high ranking for road fatalities and accidents.

Lane indiscipline is one of the major human factors accountings for 5.4% (27,431) of road accidents, 6.1% (9,201) of total deaths and 5.5% (24,628) of total injuries caused by violating traffic rules. Other elements, such as Over speeding continue to be the biggest contributing factor in road accidents in the nation, accounting for over 71% (3,19,028) of all incidents in 2019. These accidents resulted in 1,01,723 (67.3% of all fatalities) and 3,26,850 (72.4% of all injuries) fatalities and injuries, respectively. Although these factors have increased in 2019 compared to the corresponding figures of 2018, this highlights the need for stricter enforcement measures.

Balance violations such as drunk driving, jumping traffic signals, and use of mobile phones together accounted for 6% of total accidents and 8% of total deaths. When compared to those caused by human factors, traffic accidents, deaths, and injuries resulting from other causes, such as the road environment, vehicle condition, etc., accounted for 17–18% of the overall data in 2019. The majority of the causal elements point to the notion that human causes or problems with the road environment are what cause most of the road accidents in India. India is the top nation in this regard, with road accidents now being a major source of deaths and injuries worldwide and this is due to traffic rules violation.



Graph 1.1 Trends of road accidents, deaths, and injuries



Source: Road Accidents in India, 2018

Graph 1.2 Causes of Road Accidents as per 2018

The above figures make it clear that violating traffic rules is the only reason for Road accidents. In order to solve this issue, I came up with a machine learning program that will be able to detect the vehicle's number plate and save it in the blockchain. This project will make use of machine learning algorithms and models which will accurately identify rules violating vehicles by detecting the number on the number plate and store it on blockchain for further identification of the owner of the vehicle.

1.2 Problem Statement

As described in the introduction, due to the increasing number of cars in cities, there can be high quantities of traffic, which means that traffic violations are now a major issue not only in India but on a global scale. To solve the alarming problem and stop such unfathomable results like road accidents leading to injuries and deaths, systems for detecting traffic offenses and recording the facts are needed. In order to avoid penalties like fines and challans, individuals will drive safely as a result of the authorities' ability to monitor the traffic with the aid of this project. Thus, preventing traffic accidents is the result.

1.3 Objective

The project's objective is to automate the system for detecting traffic signal violations, making it simple for the authorities to monitor traffic and take prompt and effective action against the driver of the offending vehicle. Detecting, tracking, and storing details of the vehicle and their activities accurately is the main priority of the system.

1.4 Methodology

1. Machine Learning

Machine learning is a kind of artificial intelligence, by which a computer learns about a task or so without being programmed for that fully. It learns from the training data using the mapping function and tries to predict the results. The better the training data the better algorithm performs. The Machine Learning system acclimates over time and tries to learn through experience, it improves over time.

We need another tool that can identify an Indian licence plate in order to detect licence plates from images, therefore Haar cascade is employed for that purpose and is already familiar with Indian licence plates. examining the licence plate and applying some image processing: We can easily create enough details about the plate using OpenCV's grayscale, threshold, erode, dilate, and contour detection tools, as well as some parameter adjustment. From there, we can decide whether the data is valuable enough to be passed on to additional processes or not. separating the licence plate's alphanumeric characters. examining each character individually, identifying each one, joining the outcomes, and displaying the plate number as a string.

2. CNN model

CNNs, also referred to as convolutional neural networks, are deep learning neural networks created for processing structured arrays of data, including photographs. Each of the several convolutional layers that make up CNN is capable of recognising more complex shapes. A convolutional neural network is created by layering numerous invisible layers on top of one another in a specific order to create a multi-layered feed-forward neural network. In this project the following CNN layers are used:

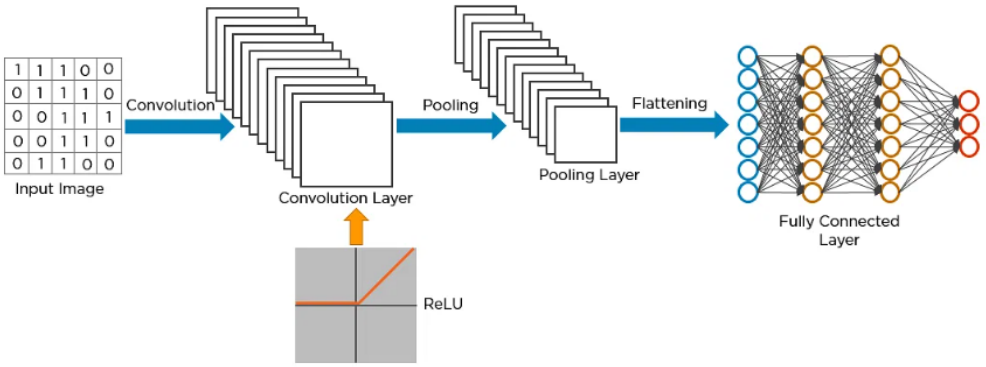
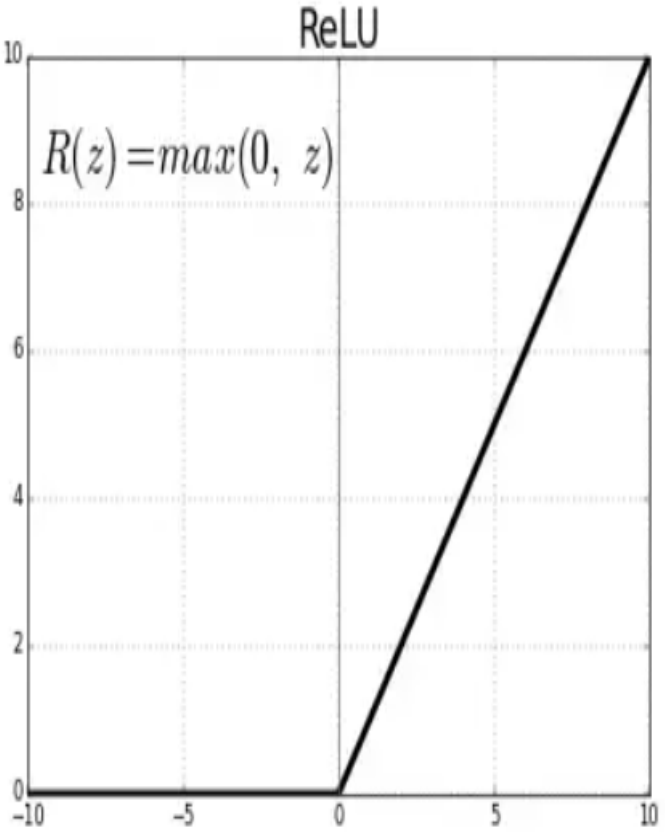


Fig. 1.1 Layers in CNN

2.1 Convolutional layer:

CNN's convolution layer serves as its main building block. It supports the lion's share of the network's computing burden. The limited region of the receptive field and the kernel, two matrices that are each composed of learnable parameters, are combined in this layer to form a dot product. The kernel has a lower surface area but is deeper than an image. This suggests that if the image has three (RGB) channels, the kernel height and width will be spatially tiny while the depth will increase to include all three channels.



$$RELU(x)=\begin{cases}0\ if\ x<0\\ x\ if\ x>=0\end{cases}$$

Graph 1.3 ‘Relu’ as activation function.

2.2 Pooling layer:

The pooling layer takes the place of the network's output in certain places by getting a summary statistic from the nearby outputs. As a result, the spatial size of the representation is decreased, which lowers the demand for computation and weights. For the pooling operation, each slice of the representation is handled separately. The most prevalent procedure, max pooling, which reports the maximum output from the neighbourhood, is utilised in this project.

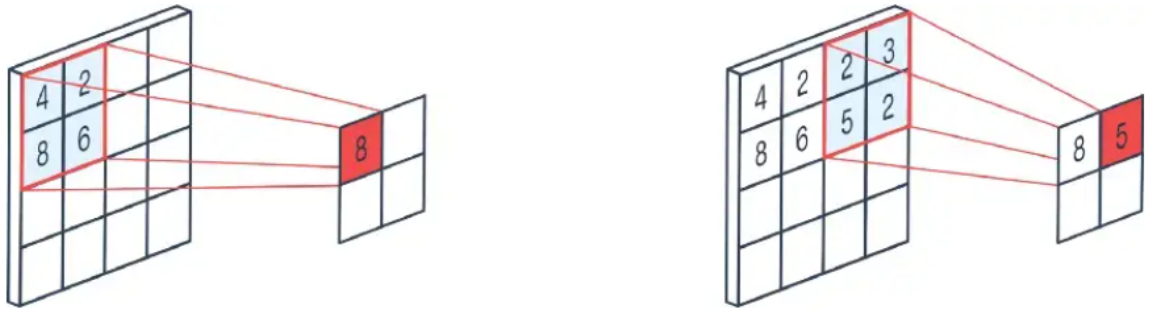


Fig. 1.2 Max pooling layer

2.3 Dropout layer:

The Dropout layer is a mask that preserves the functionality of all other neurons while removing some neurons' contributions from the subsequent layer. Because they prevent overfitting on the training data, dropout layers are essential for training CNNs. The learning of features that are only present in later samples or batches would be hampered if they were absent since the first batch of training samples has an abnormally significant impact on learning.

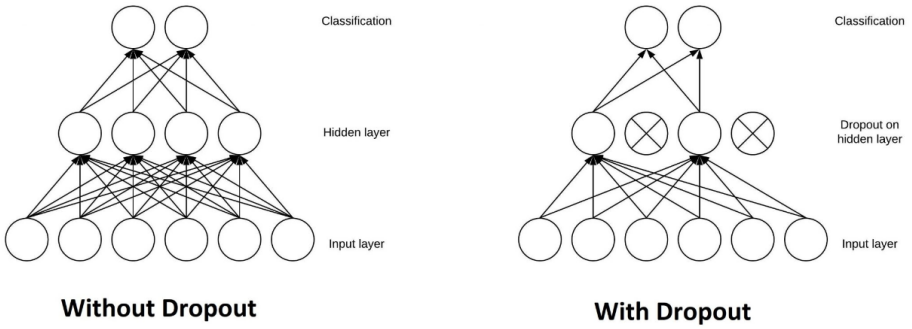


Fig. 1.3 Dropout layer

2.4 Flatten:

Convolutional layer output is in multi-dimensional shape, whereas dense layer input is in single-dimensional shape, such as a 1-D array, hence we cannot transmit convolutional layer output directly to dense layer. Therefore, between convolutional and dense layers, we'll employ the Flatten () method. All of the resulting 2-Dimensional arrays from the pooled feature maps are flattened into a solitary, extended continuous linear vector. The flattened matrix is sent as input to the fully linked layer in order to categorise the image.

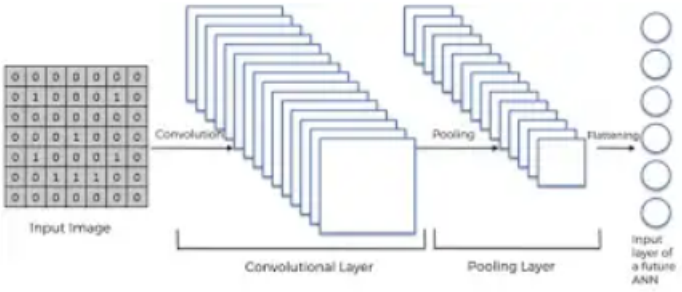


Fig. 1.4 Flatten the node data

2.5 Dense layer:

The output of the image will be provided to the dense layer after it has been processed by all convolutional and pooling layers. A basic layer of neurons is called a dense layer because every neuron in it receives input from every other neuron in the layer below it. Convolutional layer output is used to categorize images using dense layers.

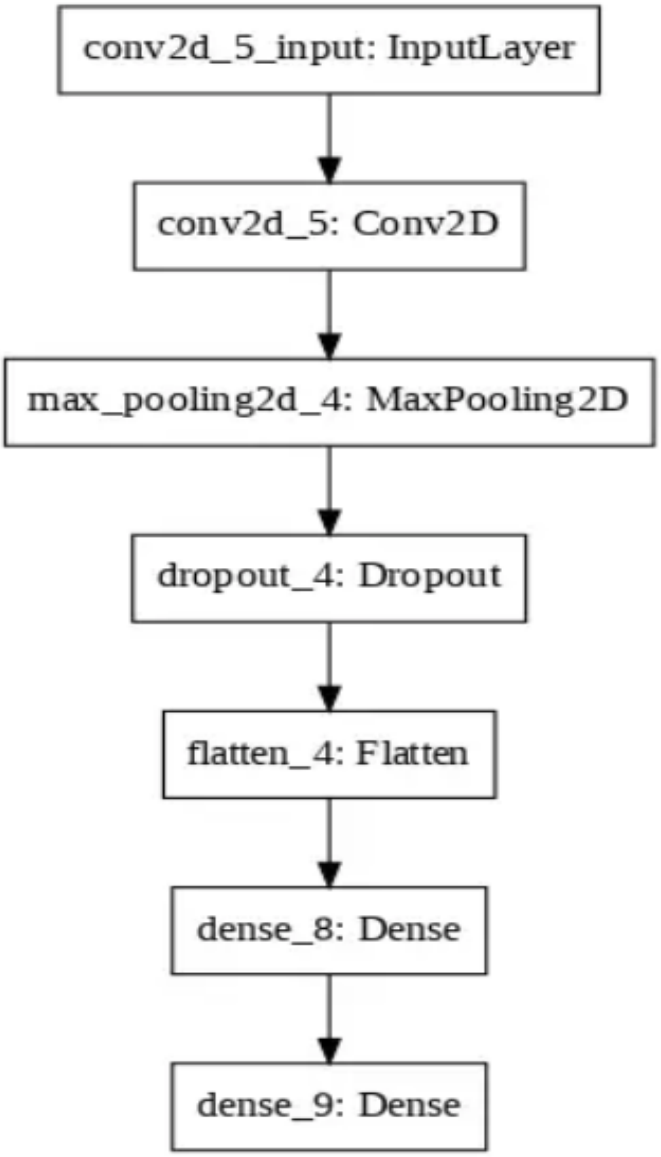


Fig. 1.5 Flow of creating a ML model and training it for characters

3. Frontend

React.js is utilized for the front end. When a vehicle's license plate is recognized up by the machine learning model, it is translated into a string and transferred to the frontend where it is then stored on a blockchain.

4. Blockchain

Utilising the Ethereum Blockchain is the project. The blockchain will store the number plate number that was determined by machine learning. In order to store the retrieved string on the blockchain, a contract was subsequently made on the remix platform. Simply defined, smart contracts are stored on the blockchain and run when specific conditions are met. They are frequently used to automate the execution of a contract so that all parties can be confident in the outcome immediately, without the need for an intermediary or further delay. It was created using the computer language Solidity.

1.5 Organization

The organization of the report is as following:

- Chapter 1 – Introduction
- Chapter 2 – Literature Survey
- Chapter 3 – System Development
- Chapter 4 – Performance Analysis
- Chapter 5 – Conclusions

CHAPTER-2: LITERATURE SURVEY

Machine learning is acknowledged as an innovative field in the detection of traffic rule violations since it has delivered high-caliber work in this area. Researchers from across the world have developed and evaluated many ways. Implementing technologies and gadgets to make the roads safer for people is fundamentally one of the strategies done to monitor traffic rule infractions and reduce the number of accidents occurring every day.

For our project work, we read through some of the research papers that gave some useful methods for number plate image recognition and blockchain storage.

1. **Md. Atikuzzaman Et al. [1]** build Vehicle Number Plate Detection and Categorization using CNN. As a result of numerous real-world implementations, they have worked on real-time vehicle number plate recognition, which has frequently been the topic of research studies. Their suggested technique is intended specifically to function on camera-captured films and allows for the real-time detection and identification of license plates. It is a unique strategy made up of three basic stages, including plate detection, segmenting class letters, and identification. They used a dataset of 5500 license plates and successfully recognized 91.38% of them at a pace of roughly 30 frames per second. With 390 test photos, they analyzed the performance of the License Plate Detection system and found that it had 96.92% accuracy, whereas Class Letter Segmentation had 94.61% accuracy with the same amount of data.

2. **N. PALANIVEL Et al. [2]** The purpose of this project was to recognize the license plate on the automobiles in challenging conditions, such as distorted, high/low light, and dirty ones. The paper

has suggested using a faster R-CNN to identify a license plate from a car using a security camera that was installed in a busy area. They developed a system that was used to record footage of the car and, for better outcomes, extract the number plate from the video using frame segmentation and picture interpolation. The optical character recognition method was applied to the resulting image for the purpose of number recognition. The proposed system is able to achieve a 99.1% accuracy to detect the number plate of the vehicle and show the vehicle's owner information.

3. **W. Li Et al. [3]** Data security challenges in connected vehicular networks are becoming increasingly problematic as connected vehicle technologies are increasingly incorporated into intelligent traffic signal management systems. This paper presented a novel decentralized and secure by design architecture based on the upcoming blockchain paradigm for connected car data security. They used this design in a simulation study to protect the Intelligent Traffic Signal System (I-SIG), a CV pilot program approved by the USDOT, from congestion attacks. They presented a blockchain-based architecture in this paper to protect intelligent traffic signal control systems from information and data attacks by converting the traditional connected vehicle network into a trustworthy and open decentralized network.

4. **Y. Yuan Et al. [4]** Blockchain can be used to create an autonomous, secure, and decentralized ITS ecosystem that makes greater use of existing ITS resources and infrastructure. This is especially useful for crowdsourcing technology. A preliminary analysis of Blockchain-based ITS (B2 ITS) was done in this paper. They described a seven-layer, ITS-focused conceptual model for blockchain and used this to solve the main B2 ITS research questions. They talked about the

connection between B2 ITS and PtMS after deciding that blockchain is one of the secure and reliable architectures for constructing the recently established parallel transportation management systems (PtMS). They concluded by presenting a case study for real-time ride-sharing services based on blockchain.

5. **C. Patel Et al. [5]** ANPR systems are widely available right now. Despite the fact that these systems are based on numerous methodologies, the task remains challenging because a number of factors, including a vehicle's high speed, non-uniform number plate, the language of the number, and changing lighting conditions, can significantly affect the overall identification rate. With these limitations, the bulk of the systems work. This study presents a wide range of ANPR approaches, taking picture size, success rate, and processing time into consideration. An examination of the ANPR's performance revealed that it had a 95.62% accuracy rate.

Table 2.1 Literature survey

Sno	Author(s)	Journal/ conference year	Published by	Methodology	Disadvantage
1	Md. Atikuzzaman Md. Asaduzzaman Md. Zahidul Islam	2019	IEEE	This paper describes the Vehicle number plate detection & categorization using CNN.	The additional RAM is necessary for training deep and intricate CNN models as well as any other exponent. With a GPU system, a large and complex CNN model will undoubtedly improve the system's overall performance. Due to the lack of various class letters, I can only recognise four different sorts of class letters.
2	N.PALANIVEL AP T.VIGNESHWARAN M.SRIVARAPPRA-DHAN R.MADHANRAJ	2020	IEEE	The paper has suggested using a faster R-CNN to identify a license plate from a car using a security camera that was installed in a busy area.	For CNN to create a convolutional feature map, the input image must be fed to it rather than the region proposals.
3	Wanxin Li Mark Nejad Rui Zhang	2020	IEEE	They used this design in a simulation study to protect the Intelligent Traffic Signal System, a CV pilot program approved by the USDOT, from congestion attacks.	Applications heavily rely on digital tokens when it comes to system architecture. This means that blockchain technology can only be used in systems that involve cryptocurrencies.

4	Rui Zhang Fei-Yue Wang	2016	IEEE	They described a seven-layer, ITS-focused conceptual model for blockchain and used this to solve the main B2 ITS research	The "Achilles' heel" of ITSs can be centralised authority or cloud-based platforms, which may be momentarily inaccessible as a result of malicious attacks, performance restrictions, or just faulty operations.
5	Chirag Patel Dipti Shah Atul Patel	2013	IEEE	Using picture size, success rate, and processing time as factors, many ANPR techniques are presented in this paper.	The majority of other systems use online databases to pull offline photos of vehicles, therefore the actual results of ANPR may differ from the results.

CHAPTER-3: SYSTEM DEVELOPMENT

3.1 Model Development

- **Tools Platforms/Technology/Languages Used in number plate detection**

  - **Programming Language:** Python 3.7

  - **IDE:** PyCharm

  - **Environment:** Anaconda

  - **OpenCV**

    An open-source library of computer vision and machine learning techniques called OpenCV (Open Source Computer Vision) was created to assist programmers in developing sophisticated image processing applications. It was initially created by Intel, and the OpenCV team currently looks after it. A sizable open-source toolbox for computer vision, machine learning, and image processing is called OpenCV. Python and C++ are only a couple of the many programming languages that OpenCV supports. It can analyze photos and movies to recognize items, faces, and even a person's handwriting. Real-time computer vision is the primary focus of the OpenCV library of programming functions, which is also open-source.

    Image processing is the technique of using software to analyze, enhance, or modify digital images. It is an important field of computer

science that has a wide range of applications, including medical imaging, robotics, and video processing.

OpenCV provides a variety of image processing functions and algorithms that can be used for tasks such as image filtering, edge detection, feature extraction, object recognition, and more. It supports various programming languages such as C++, Python, and Java.

Some of the key features of OpenCV include support for multiple platforms, real-time processing, GPU acceleration, and a large community of developers and users. It is widely used in industry and academia for research and development of computer vision applications.

In summary, OpenCV is a powerful tool for image processing that provides a range of functions and algorithms for developers to create advanced computer vision applications.

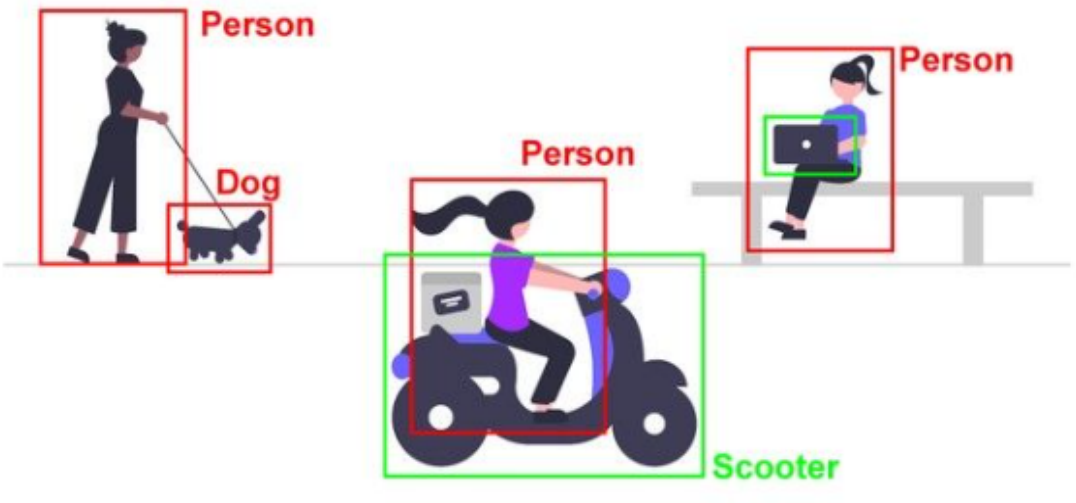


Fig. 3.1 Example of Opencv

- **Haar cascade**

In order to recognise items in an image or video, a machine learning object detection technique is used. The Haar cascade technique allows for the detection of objects regardless of their size or location in the image. This algorithm is simple and real-time capable. We can program a haar-cascade detector to recognize a variety of items, including vehicles, bikes, structures, fruits, etc. The Haar cascade employs the cascading window and seeks to calculate features in each window in order to determine whether it might be an item. As a classifier, Haar cascade is used. Positive data points are those that are a part of the object we have spotted, while negative data points are those that do not. Haar cascades are quick and effective in real-time. But modern methods of object detection are more accurate than Haar cascade.

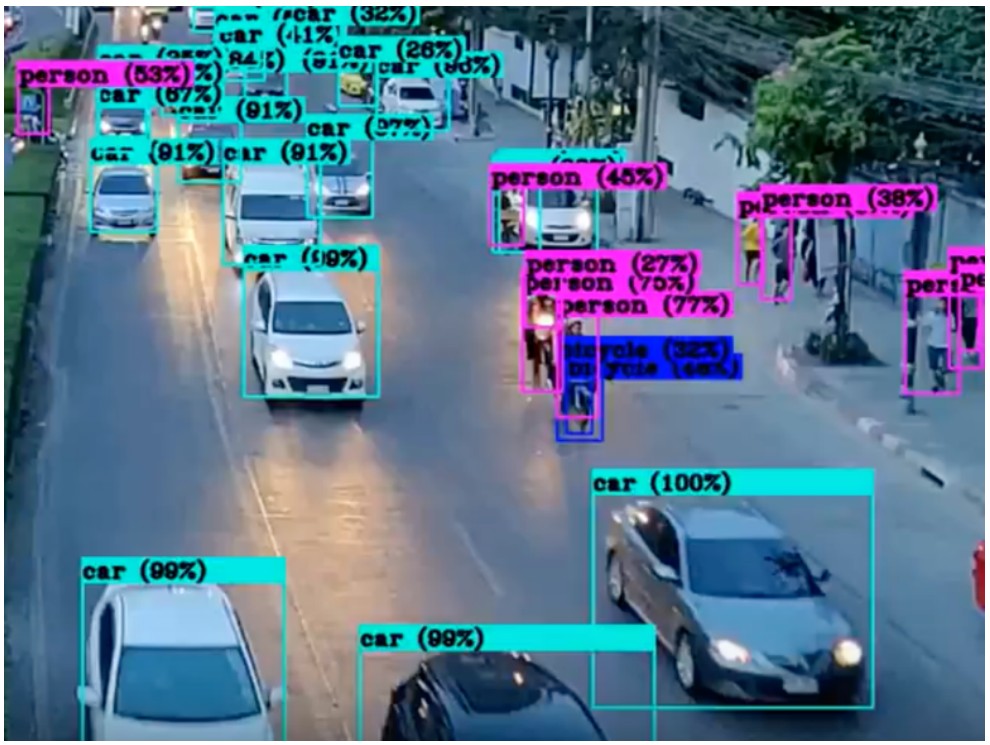


Fig. 3.2 Example of Haar Cascade

- **TensorFlow**

TensorFlow is an open-source library for doing numerical computation and large-scale machine learning. TensorFlow gathers a number of machine learning and deep learning models and algorithms and makes them practical through the usage of typical programmatic metaphors. High-performance C++ is utilised to run such programmes, and it provides a simple front-end API for building apps using Python or JavaScript. Deep neural networks, such as word embeddings, recurrent neural networks, sequence-to-sequence models for machine translation, natural language processing, and PDE-based simulations, can be trained and used for a variety of tasks with the aid of TensorFlow. The nicest thing is that TensorFlow offers production prediction at scale using the same models that were used for training. Furthermore, TensorFlow offers a vast library of pre-trained models.

- **Keras**

In order to create and assess deep learning models, Keras is a robust and user-friendly free open-source Python framework. Deep learning is about as simple as deep learning can be with Keras, which is user-friendly and has a large community of supporters. Google designed the high-level Keras deep learning API to implement neural networks. It is designed in Python and intended to make neural network implementation simpler. Multiple neural network backend computation is supported in addition. Keras is fairly easy to comprehend and use since it offers a high level of abstraction Python frontend with the option of several different back-ends for computation. Keras is slower than other deep learning frameworks as a result of this. TensorFlow has approved Keras as its official high-level

API. Keras can be used to complete deep learning quickly because it comes with built-in modules for all neural network functions. TensorFlow incorporates Keras.

- **Scikit-Learn**

It is a Python programming language machine learning package that is available for free. Scikit-learn is probably the most useful Python machine learning library. The sklearn toolkit includes a number of efficient machine learning and statistical modelling methods, including classification, regression, clustering, and dimensionality reduction. Machine learning models are created using sklearn. Scikit-learn has a ton of capabilities, but they shouldn't be used for reading, modifying, or summarizing the data.
Scikit-Learn, also known as sklearn, is a popular Python library for machine learning. It is built on top of NumPy, SciPy, and Matplotlib, and provides a wide range of tools for data analysis and machine learning tasks. Here are some key features of Scikit-Learn:

- Consistency: Scikit-Learn provides a consistent API, making it easy to switch between different machine learning models and algorithms.

- Broad range of algorithms: Scikit-Learn offers a range of algorithms for classification, regression, clustering, and dimensionality reduction, including popular algorithms like decision trees, random forests, and support vector machines.

- Data preprocessing: Scikit-Learn includes tools for data preprocessing, including feature scaling, feature extraction, and data normalization.

- Model selection and evaluation: Scikit-Learn provides tools for selecting the best model for a given dataset, as well as for evaluating the performance of different models.

- Integration with other libraries: Scikit-Learn can be easily integrated with other Python libraries, including Pandas for data manipulation and Matplotlib for data visualization.

Overall, Scikit-Learn is a powerful and easy-to-use tool for machine learning in Python, with a large and active community that supports its development and use.

- **Numpy**

Numpy is one of the most well-known Python packages for scientific computing. It provides a multidimensional array object in addition to variations like matrices and masks that can be used in a variety of mathematical functions. Numerous other popular Python packages, including pandas and matplotlib, depend on Numpy and are compatible with it. Numpy is a general-purpose array processing package. It provides an extremely quick multidimensional array object along with the ability to interact with these arrays. The core Python module for scientific computing is this one. Numpy is a potent multi-dimensional data container in addition to its obvious uses in science.

Large, multi-dimensional arrays and matrices can be supported using the Python module NumPy, which also offers support for a large number of mathematical operations.

Here are some key features of NumPy:

- Multi-dimensional arrays: NumPy provides support for multi-dimensional arrays, which can be used to represent matrices, images, and other data structures.

- Broadcasting: NumPy supports broadcasting, which allows for efficient element-wise operations on arrays of different sizes and shapes.

- Mathematical functions: NumPy provides a wide range of mathematical functions, including functions for linear algebra, Fourier analysis, and random number generation.

- Integration with other libraries: NumPy can be easily integrated with other Python libraries, including SciPy for scientific computing and Matplotlib for data visualization.

- Efficient performance: NumPy is optimized for efficient performance, with support for multi-core and multi-processor systems, making it suitable for large-scale data processing and scientific computing tasks.

Overall, NumPy is a powerful and essential tool for scientific computing and data analysis in Python, with a large and active community that supports its development and use.

- **Matplotlib**

Matplotlib is a cross-platform library for creating graphical charts and data visualisation for Python and its numerical extension NumPy. As a result, it offers a reliable open-source alternative to MATLAB. Developers can include graphs in GUI programmes by using the matplotlib APIs (Application Programming Interfaces). The majority of the time, a visual data plot may be created using just a few lines of code thanks to the way Python matplotlib scripts are constructed.

To produce excellent graphs and plots, many people utilise the Python data visualisation tool Matplotlib. It has a number of functions for making various types of plots, including line plots, scatter plots, bar plots, and histograms.
Here are some key features of Matplotlib:

- Compatibility: Matplotlib works well with NumPy arrays and other libraries such as Pandas, SciPy, and Scikit-learn.

- Versatility: Matplotlib offers a range of customization options, including different color schemes, line styles, and marker styles, making it suitable for a wide range of visualizations.

- Object-oriented interface: Matplotlib uses an object-oriented API, which allows for greater flexibility and customization in creating plots.

- Interactive visualization: Matplotlib can be used to create interactive plots, including zooming and panning functionality.

- 3D visualization: Matplotlib also provides tools for creating 3D plots and animations.

Overall, Matplotlib is a powerful and flexible tool for data visualization in Python, with a large and active community that supports its development and use.

- **FastAPI**

Python APIs can be built using FastAPI, a contemporary, quick (high-performance), web framework.

Key characteristics include:

1) Quick: Very strong performance, comparable to Go and NodeJS. among the swiftest Python frameworks on offer.
2) Quick to code: Double or triple the rate at which features are developed.
3) Fewer bugs: About 40% fewer mistakes caused by developers.
4) Intuitive: Excellent editor assistance. Finished work everywhere. Debugging takes less time.
5) Simple: Created to be simple to use and learn. less time reading documents.
6) In brief: Reduce code duplication. From each parameter declaration, multiple features. less bugs.
7) Robust: Obtain code fit for production. with interactive documentation that is automatic.
8) Standards-based: Built on (and completely compatible with) the open API standards.

3.2 Tools Platforms/Technology/Languages Used in frontend

- **Code Editor:** VsCode

- **HTML (Hypertext Markup Language)**

The language used to describe the structure of Web pages is HTML.

1) Publish online documents with headings, text, tables, lists, images, etc. thanks to HTML.

2) Quickly and easily access internet information by clicking on hypertext links.

3) Create forms for using online services to make reservations, place orders for goods, and more.

4) Directly integrate spreadsheets, audio, video, and other apps into their documents. In this project few basic concepts of HTML are used.

The markup language HTML (Hypertext Markup Language) is used to write and organise material for web pages and other sorts of documents that can be viewed in a web browser. online development relies heavily on HTML, which is used to specify the organisational framework for both online pages and applications.

The structure and content of a document are defined by a system of tags in HTML. Different parts of the document, including headings, paragraphs, images, and links, are marked up using these tags. Developers can design a hierarchical

hierarchy of material that is simple to read and browse by making use of these tags.

Some of the key features of HTML include:

- Semantics: HTML includes semantic tags that describe the meaning and purpose of different elements, which improves accessibility and search engine optimization (SEO).
- Hyperlinks: HTML allows for the creation of hyperlinks, which enable users to navigate between different pages and resources on the web.
- Forms: HTML includes form elements that enable users to submit data to a web server, which is useful for collecting user input and processing it on the server side.
- Multimedia: HTML includes tags for embedding multimedia content such as images, audio, and video in web pages.

HTML is often used in conjunction with other web development technologies such as CSS (Cascading Style Sheets) and JavaScript to create dynamic and interactive web experiences.

In summary, HTML is a markup language used to create and structure content for web pages and other types of documents that can be displayed in a web browser. It uses a system of tags to define the structure and content of a document, and has become a core technology used in web development.

- **CSS (Cascading Style Sheets)**

The term "CSS" refers to the language used to describe how Web pages are presented, including their colours, layout, and fonts. It makes it possible to modify the presentation to work on a variety of screens, including printers, small screens, and enormous screens. It is not necessary to use HTML in order to use CSS with any XML-based markup language. Because HTML and CSS are distinct from one another, it is easier to maintain websites, share style sheets across pages, and modify pages for different contexts.

The styling language CSS (Cascading Style Sheets) is used to dictate how HTML (Hypertext Markup Language) is presented and other types of documents. It is used to create visually appealing and user-friendly web pages and applications.

CSS works by defining styles for HTML elements such as text, headings, images, and other elements. These styles include properties such as font size, color, spacing, and positioning, among others. By defining these styles, CSS allows developers to control the layout and appearance of web pages and applications.

CSS code is typically written in separate files or within the HTML code itself. This separation of style from content allows for easier maintenance and updates of web pages and applications.

Some of the key features of CSS include:

- Cascading: Styles can be defined at different levels and cascaded down to affect multiple elements on a page.
- Selectors: CSS uses selectors to target specific HTML elements and apply styles to them.
- Box model: CSS includes a box model that defines how elements are positioned and sized on a web page.
- Responsive design: CSS allows for the creation of responsive designs that adapt to different screen sizes and devices.

CSS has become an essential part of web development, as it enables developers to create visually appealing and user-friendly web pages and applications. It is widely used in conjunction with HTML and JavaScript to create dynamic and interactive web experiences.

In summary, CSS is a styling language used to describe the presentation of HTML and other types of documents. It allows developers to control the layout and appearance of web pages and applications, and is an essential part of modern web development.

o REACT.JS

A JavaScript framework and library available as open source is called React.js. With much less code than JavaScript, it is used

to create interactive user interfaces and web applications quickly and effectively. Reusable component creation in React allows for the development of applications. These parts make up a final interface piece by piece, which when put together creates the full application's user interface. The main responsibility of React in an application is to manage the view layer by offering the finest and most effective rendering execution. React.js encourages developers to break down these sophisticated user interfaces into distinct reusable components that serve as the foundation for the entire UI rather than treating the entire UI as a single entity.

ReactJS (or simply React) is an open-source JavaScript library developed by Facebook for building user interfaces (UIs) for web applications. React enables developers to create reusable UI components and manage the state of the application in an efficient and easy-to-understand way.

React is based on the concept of components, which are modular and reusable pieces of code that define how a part of a UI should be rendered. Components can be combined to create complex UIs, and changes to the state of the application are automatically reflected in the UI.

Some of the key features of React include:

- Virtual DOM: React uses a virtual representation of the DOM to improve performance by minimizing updates to the actual DOM.
- JSX: React allows developers to use JSX (a syntax extension to JavaScript) to write HTML-like code within JavaScript files.

- One-way data binding: React uses one-way data binding, which means that changes to the state of the application are automatically reflected in the UI, but not vice versa.
- Server-side rendering: React can be used for server-side rendering, which allows for faster initial loading times and improved search engine optimization (SEO).

React has become a popular choice for building UIs for web applications due to its flexibility, performance, and large community of developers. It is often used in conjunction with other web development technologies, such as Redux for managing application state and React Native for building mobile applications.

In summary, React is a JavaScript library for building user interfaces for web applications. It is based on the concept of components, uses a virtual DOM for performance optimization, and has become a popular choice for building UIs due to its flexibility, performance, and large community of developers.

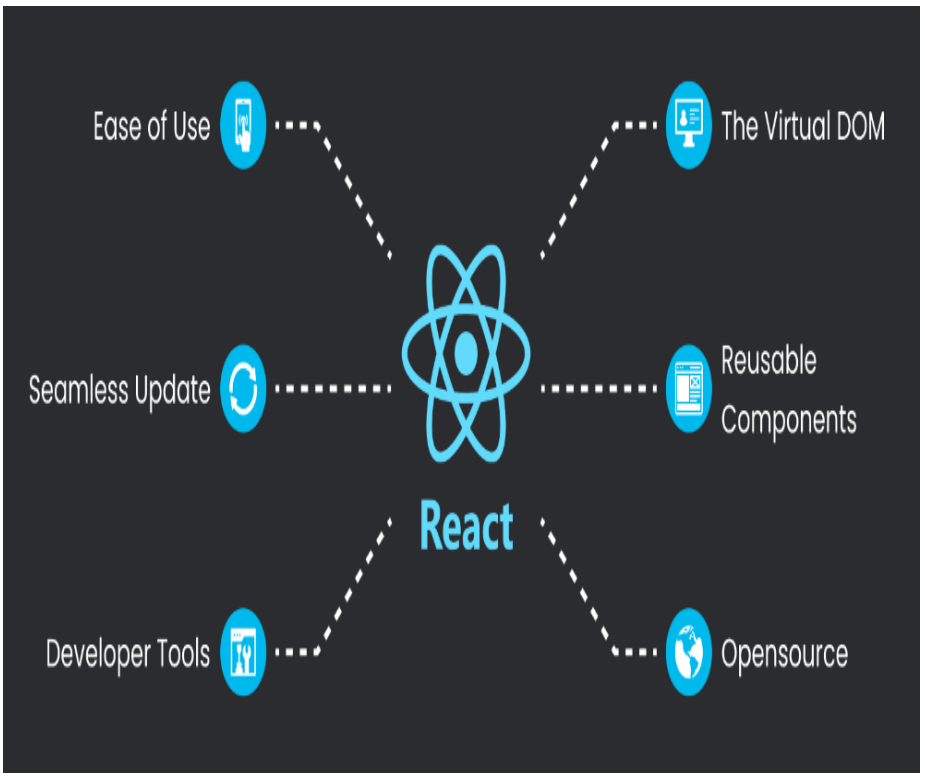


Fig. 3.3 Characteristics of React.JS

3.3 Tools Platforms/Technology/Languages Used in Blockchain

- **IDE:** Remix Ethereum

- **Programming language:** Solidity

  Solidity is an object-oriented programming language. For the purpose of constructing smart contracts on different blockchain systems, such as Ethereum. For experienced web developers, Solidity's syntax is similar to that of ECMAScript. It enables complicated member variables for contracts, such as structs and maps with arbitrary hierarchies. Inheritance is supported by solidity contracts, including multiple inheritance with C3 linearization.

  The Ethereum blockchain uses the high-level programming language Solidity to create smart contracts. It was created by the Ethereum Foundation, and a sizable developer community now looks after it.

  In smart contracts, the terms of the agreement between the buyer and seller are directly written into lines of code. These contracts self-execute. They enable the development of decentralised applications (dApps), including cryptocurrencies, decentralised exchanges, and more, on the Ethereum network.

  Solidity is designed to be similar to JavaScript and has a syntax that is easy to read and write. It supports object-oriented programming, inheritance, and other features commonly found in programming languages.

Some of the key features of Solidity include:

- Strong typing and type inference
- Support for inheritance and interfaces
- Libraries and reusable contracts
- Event logging and error handling
- Integration with Ethereum Virtual Machine (EVM)

Bytecode created from Solidity code can be run on the Ethereum Virtual Machine (EVM). The bytecode is then published as a smart contract on the Ethereum network.

A high-level programming language called Solidity is used to create smart contracts for the Ethereum network. It contains a number of features that make it simple to build and deploy smart contracts on the Ethereum network and is built to be similar to JavaScript.

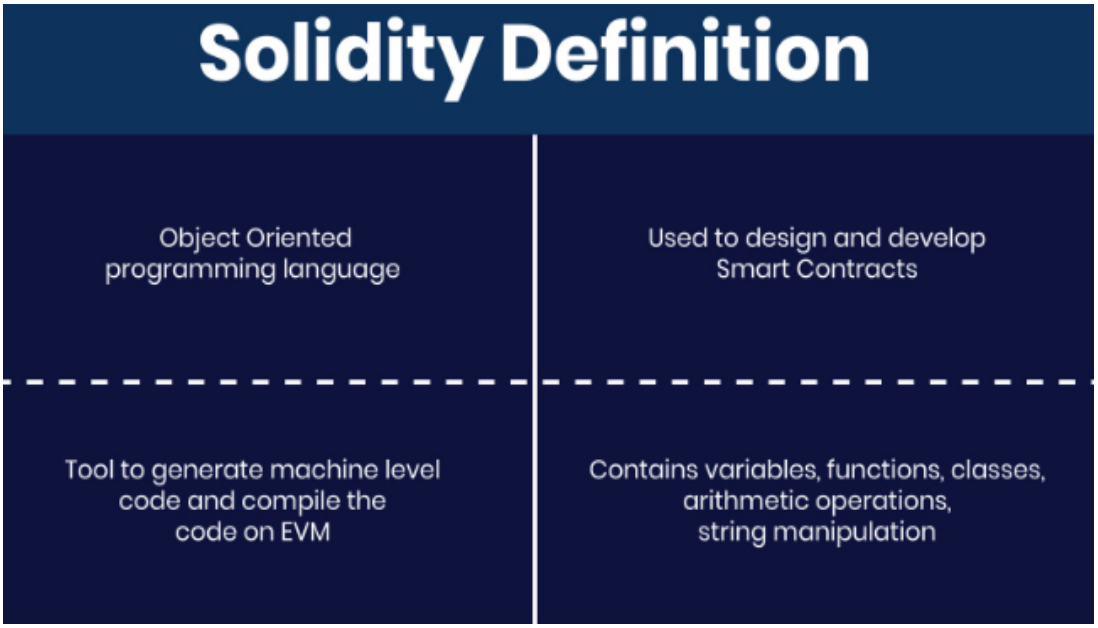


Fig. 3.4 Solidity Definition

- **Smart Contracts**

  In line with the terms of a contract or other agreement, a "smart contract" is an electronic programme or transaction protocol that is created to automatically carry out, manage, or record legally significant events and actions. The objectives of smart contracts are to minimise intentional and unintentional exceptions as well as to reduce the need for dependable middlemen, arbitration costs, and fraud damages. Smart contracts and cryptocurrencies are often related, and Ethereum's smart contracts are often recognised as a fundamental component of decentralised inance (DeFi) and NFT applications.

  By executing code that is kept on a blockchain, smart contracts function. The contract automatically executes and performs the obligations thereunder when certain criteria are satisfied, such as when a payment is received or a certain date is reached.

  Smart contracts' transparency and immutability are two of its main advantages. A smart contract cannot be changed or removed after it is published to a blockchain. This makes sure that there is no chance of fraud or other forms of manipulation and that everyone involved in the contract understands its contents.

  Smart contracts are being used in a variety of industries and applications, including:

  - Cryptocurrencies and digital assets
  - Decentralized finance (DeFi) applications, such as lending and borrowing platforms
  - Supply chain management and logistics

- Real estate transactions and property management
- Voting and governance systems

While smart contracts could revolutionize a number of industries, there are also some challenges associated with their use. For example, the complexity of smart contract code can make it difficult to identify and fix bugs, and the lack of a central authority can make dispute resolution more challenging.

In summary, Self-executing computer programmes known as smart contracts automatically enforce the terms of a contract between two parties. They are transparent, immutable, and have a range of applications in various industries.

o MetaMask Software

The Ethereum blockchain can be interacted with using the software wallet MetaMask. It enables users to interact with decentralized applications by giving them access to their Ethereum wallet via a browser extension or mobile application. Users may safely connect to decentralized applications through a compatible web browser or the mobile app's built-in browser, save and manage account keys, broadcast transactions, send and receive A user's MetaMask wallet can be connected to, authenticated, and integrated with other smart contract features by websites or other decentralized applications using JavaScript code. This enables the website to send the user action prompts, signature requests, or transaction requests through MetaMask as an intermediary.

MetaMask is a browser extension and software wallet that allows users to interact with decentralized applications (dApps) on the Ethereum blockchain. It is available as a Chrome, Firefox, and Brave browser extension, and as a mobile app for iOS and Android devices.

MetaMask allows users to create and manage Ethereum accounts, store and send Ether (ETH) and other ERC-20 tokens, and interact with dApps without the need for a full Ethereum node. It provides a user-friendly interface for interacting with the Ethereum blockchain and eliminates the need for users to manually input transaction details such as gas limits and gas prices.

MetaMask also provides users with enhanced security features, such as a seed phrase backup and password protection for their accounts. It also allows users to connect to various Ethereum networks, such as the Ethereum mainnet, testnets, and private networks.

In addition to its wallet functionality, MetaMask also provides a developer API that allows developers to integrate MetaMask with their dApps and enable users to seamlessly interact with their applications.

MetaMask has become a popular choice for Ethereum users due to its user-friendly interface, enhanced security features, and developer-friendly API. It has played a significant role in increasing the adoption of Ethereum and dApps by making it easier for users to interact with the Ethereum blockchain.

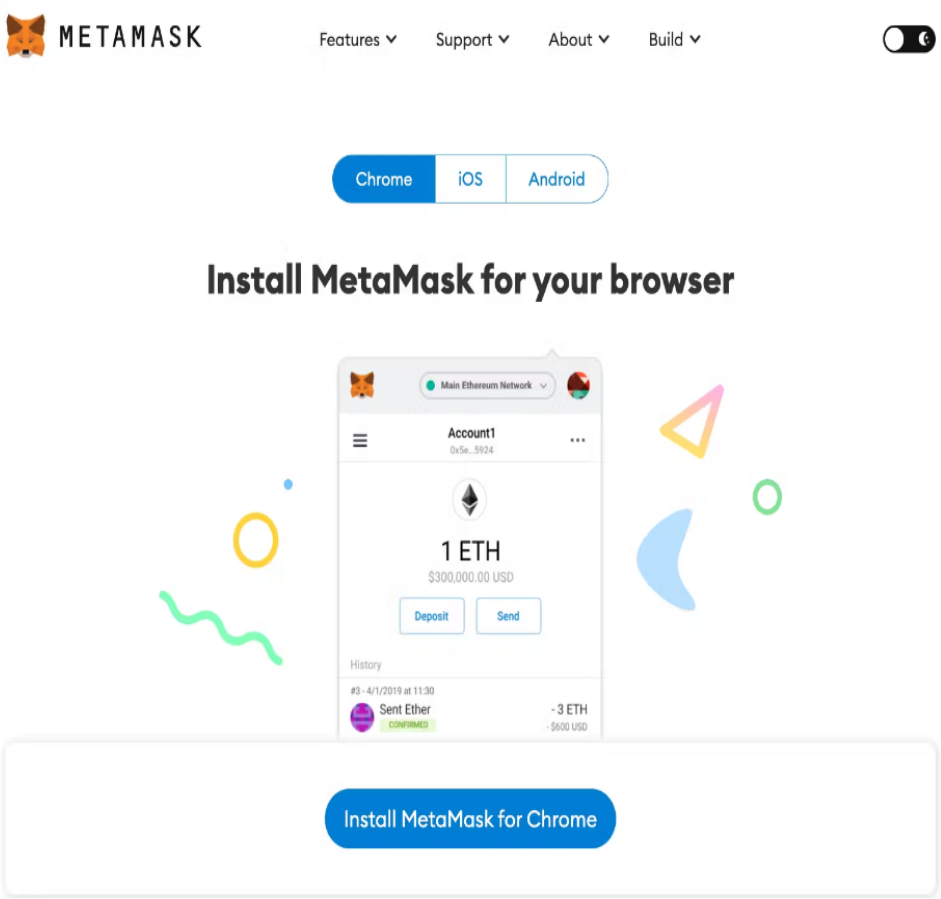
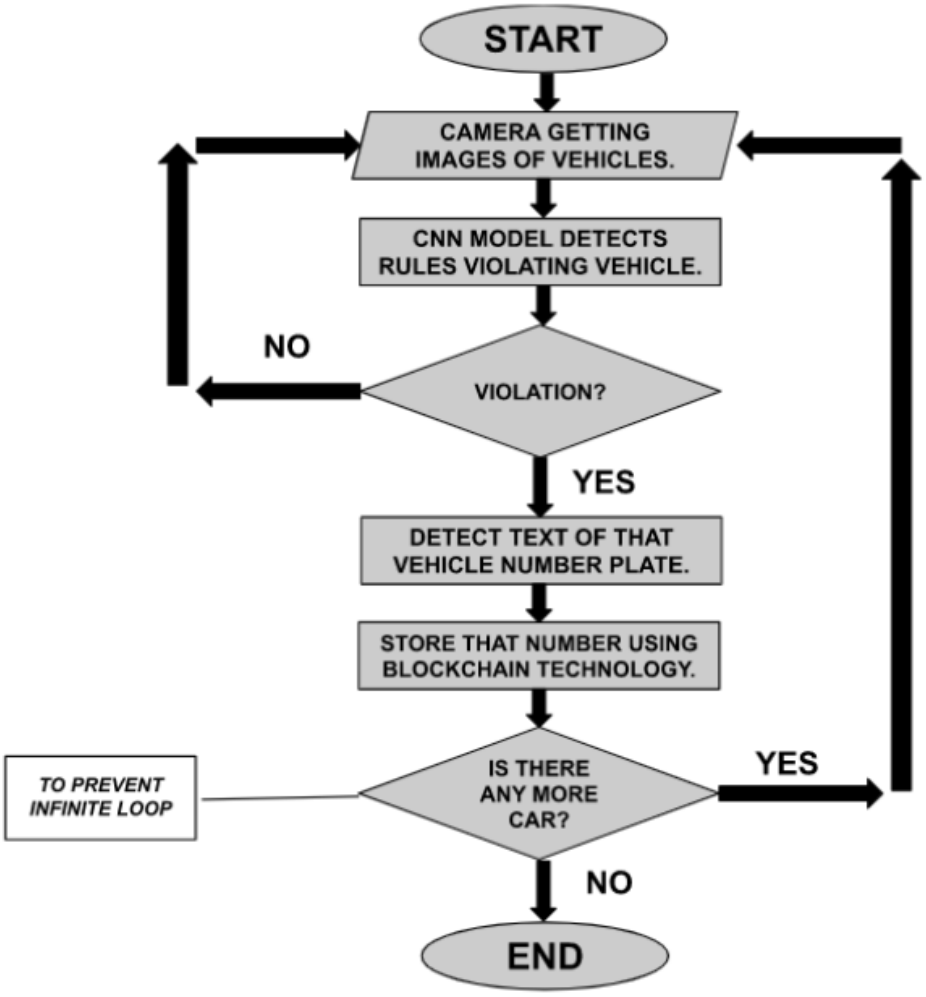


Fig. 3.5 MetaMask Software

3.4 Algorithm/ Design of the project

In this section, we summarized our project's working in a flowchart. It progresses from detection to image processing, questioning violation to storing of number plate details in blockchain and finally displaying and stopping the code.

Below is the flowchart:



Flowchart. 3.1 Algorithm/ Design of the project

3.5 Implementation

- **Setting up the environment for image detection**

  In this part of code, we first import all the libraries and functions needed for the implementation. We import:

  - matplotlib.pyplot
  - numpy
  - cv2
  - tensorflow
  - tensorflow.keras.backend
  - os.path
  - from sklearn.metrics import f1_score
  - from tensorflow.keras import optimizers
  - from tensorflow.keras.models import Sequential
  - from tensorflow.keras.preprocessing.image import ImageDataGenerator
  - from tensorflow.keras.layers import Dense, Flatten, MaxPooling2D, Dropout, Conv2D
  - from tensorflow.keras.models import load_model
  - from fastapi import FastAPI

- **Number plate detection**

  In this part of code, we create a function called "detect_plate(img)". This function

  - Detects and performs blurring on the number plate.
  - Then, it detects number plates and returns the coordinates and dimensions of detected license plate's contours.
  - After that, it extracts the Region of Interest of the license plate for blurring.

- Finally, represent the detected contours by drawing rectangles around the edges.
- Returns the processed image.

- **Performing image processing**

  In this part of code, we create a function called "segment_characters(img)". This function

  - Preprocess cropped license plate image.
  - Make borders white
  - Eliminates the characters contours sizes of cropped license plates
  - Get contours within a cropped license plate.
  - Return characters in a string.

- **Segmenting the alphanumeric characters from the license plate.**

  In this part of code, we create a function called "find_contours(dimensions, img)". This function

  - Finds all contours in the image
  - Retrieve potential dimensions
  - Check largest 5 or 15 contours for license plate or character respectively
  - Detects contours in binary image and returns the coordinates of rectangle enclosing it
  - Checking the dimensions of the contours to filter out the characters by contour's size
  - Stores the x coordinates of the character's contour, to used later for indexing the contours

- Extracting each characters using the enclosing rectangle's coordinates
- Make result formatted for classification: invert colors
- List that stores the character's binary image (unsorted)
- Return characters an ascending order with respect to the x-coordinates (most-left character first)
- Arbitrary function that stores sorted list of character indices
- Stores character images according to their indices.

o **Creating a Machine Learning model and training it for characters**

In this part of code, we create a Machine Learning model to be trained. We validate data.

- We add layers:
    - Convolutional 2D layers with activation 'Relu' and 'softmax'
    - MaxPooling2D layer with pool_size = (4,4)
    - Dropout(0,4)
    - Flatten()
    - Dense with activation 'Relu' and 'softmax'
- Compiler
    - Loss = 'sparse_categorical_crossentropy
    - Optimizer = optimizers.Adam(lr=0.0001)
    - Metrics = [custom_f1score]
- Define epochs = 80, verbose = 1.

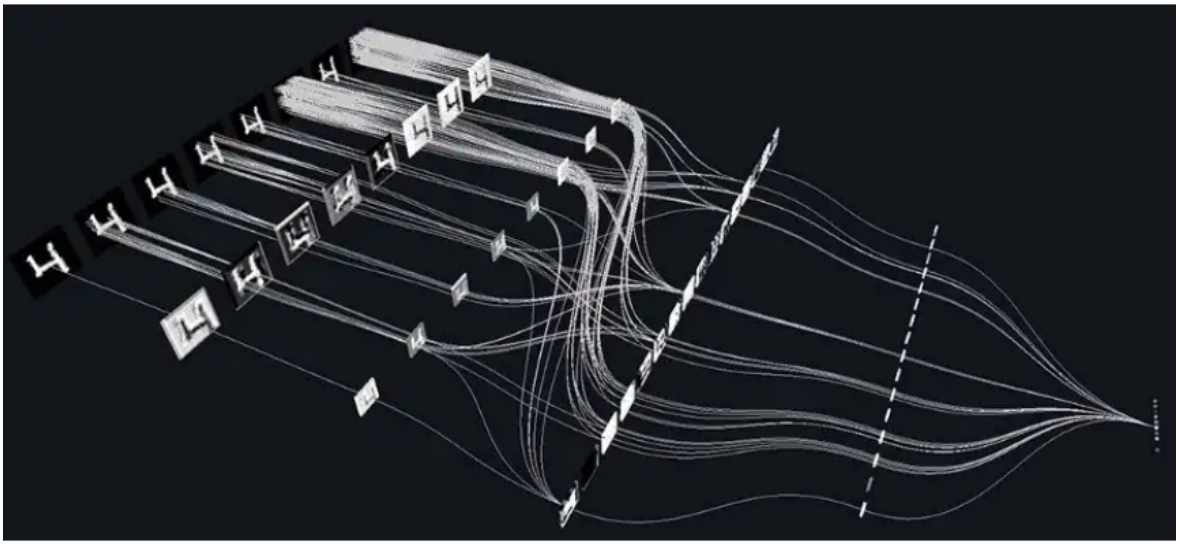


Fig. 3.6 Create a Neural Network

- **Training CNN Model**

In this part of the code, we train our model.

- All image will be resized to 28X28
- We use the target directory and store all resized images.
- Data we will be using contains images of alphabets (A-Z) and digits (0–9)
- train test split of 80:20
- We call the ImageDataGenerator function with parameters: rescale = 1. /255, width_shift_range = 0.1, height_shift_range = 0.1.
- We use train_datagen.flow_from_directory.
- Width shift: Accepts a float value denoting by what fraction the image will be shifted left and right.
  Height shift: Accepts a float value denoting by what fraction the image will be shifted up and down.
- We will use ‘categorical_crossentropy’ as loss function, ‘Adam’ as optimization function and ‘Accuracy’ as our error matrix.

As shown in the given image, first download the train dataset and the test dataset, then separate them into "train" and "test" folders on your computer. The train folder should have 'n' folders, each holding pictures from a different class. Make a validation set by selecting images from the train folder and moving them to a new folder called "valid." Creating a validation set is frequently a laborious process. You could utilize them rather than making them manually if the validation set is already available. Additionally, the test folder ought to have a single folder that houses all of the test images. It is crucial to give your class folders names that correspond to the labels inside, as this will make things simpler for you in the future.

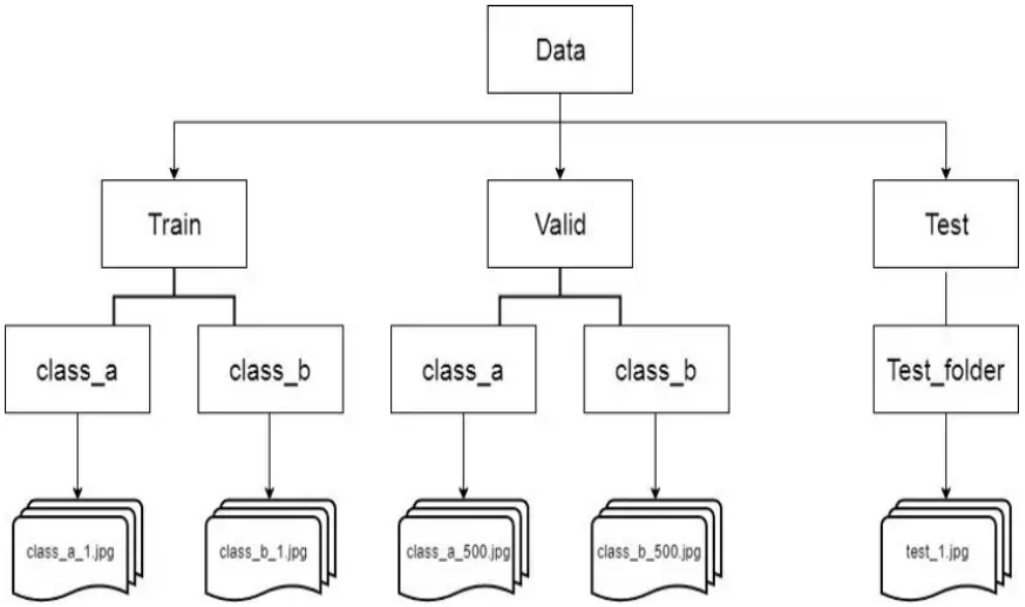


Fig. 3.7 The directory structure for binary classification problem

Frontend development

o CSS development

Now we get to the Frontend part of the code. We use CSS for designing the frontend to display details.

- In the body, we mention margin: 0, font-family, -web-font-smoothing, -moz-osx-font-smoothing.
- In the code part, we mention font-family.

o Index creation

This part of the code is in source.

- It imports:
  - React from ‘react’
  - ReactDOM from ‘react-dom/client’
  - ‘./index.css’
  - App from ‘./App’
- const root = ReactDOM.createRoot(document.getElementById(‘root’))
- Renders root

o JavaScript Implementation

Now after we have designed the display, we get to the working part of the display. We use JavaScript for working on the frontend to display details.

- It import
  - {useState, useEffect} from ‘react’
  - {ethers, util} from “ethers”
  - abi from “./contracts/numberplate.json”
- Functions:
  - App = ()

- checkIfWalletIsConnected = async ()
- getVoilatersNumber = async ()
- setVoilatersNumber = async (event)

Blockchain development

o Creating contract

Now we get to the blockchain part of the code. We use Solidity for this.

- Set pragma solidity >=0.7.0 <0.9.0
- contract NumberPlate
- Functions:
    - store(string)
    - retrieve()

The following image shows the activity and recent transactions done in metamask.

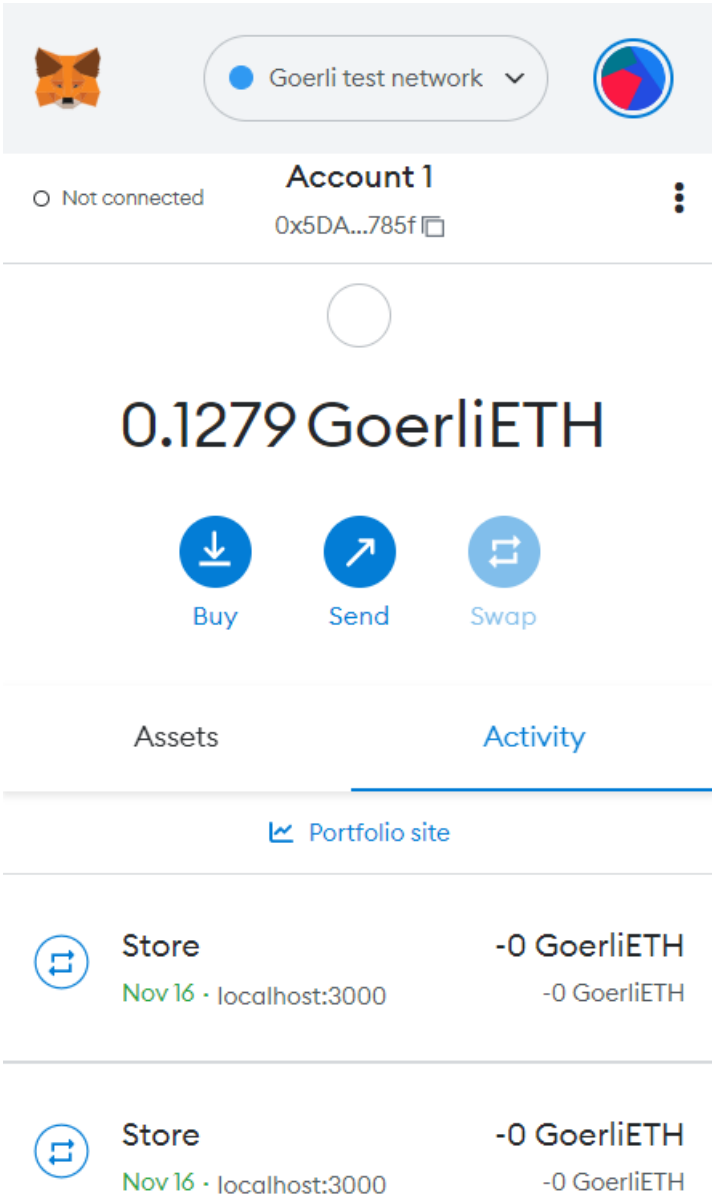


Fig. 3.8 MetaMask interface

The following image shows the transaction completed in the metamask.

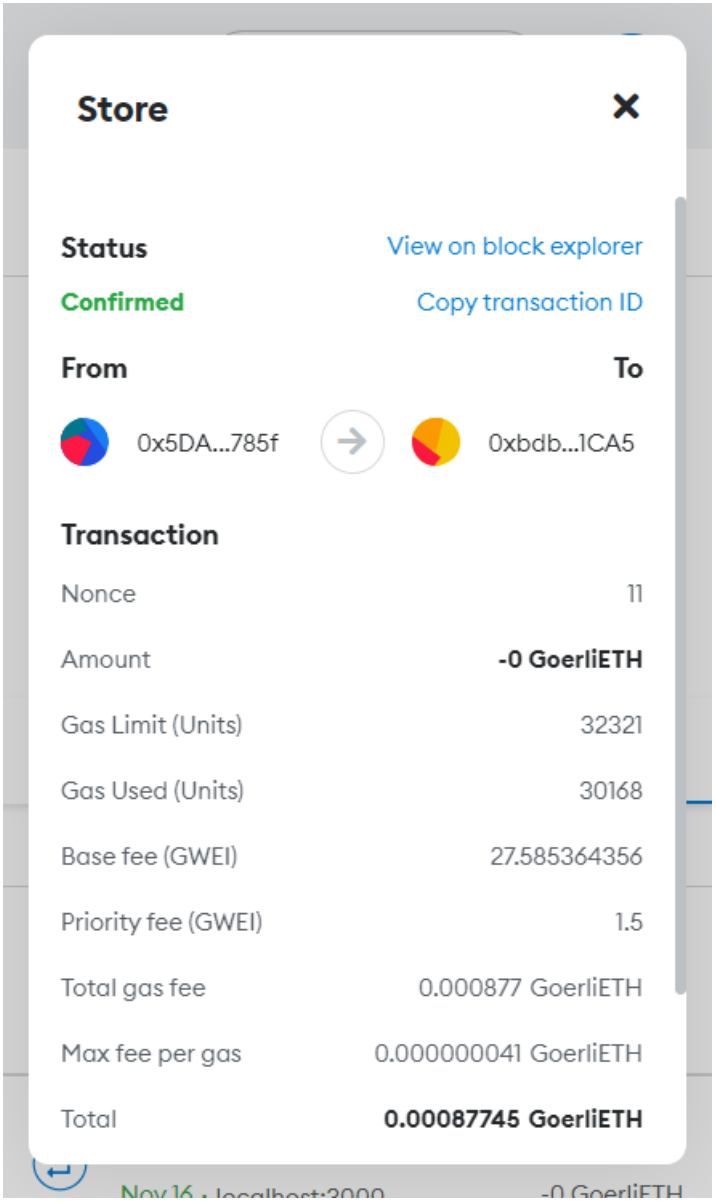


Fig. 3.9 Transactions MetaMask

Chapter-4: PERFORMANCE ANALYSIS

4.1 Results

o Results of image detection

□ Input image

About the image:

Dimensions: 4160*3120

Width: 4160 pixels

Horizontal resolution: 72 dpi

Vertical resolution: 72 dpi

The input image is shown below:

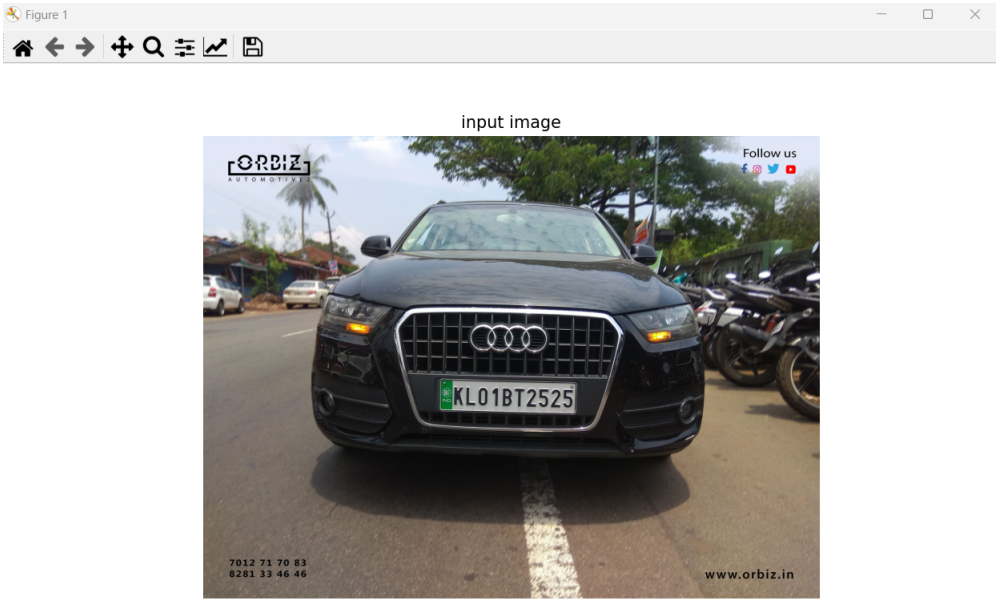


Fig. 4.1 Input image

□ Detected Image

detected license plate in the input image

Fig. 4.2 Detected number plate from the image

□ Extracted number plate

Fig. 4.3 Extraction

□ The binary images of 10 extracted characters

Fig. 4.4 Binary Conversion

□ Plate with contours drawn in blue

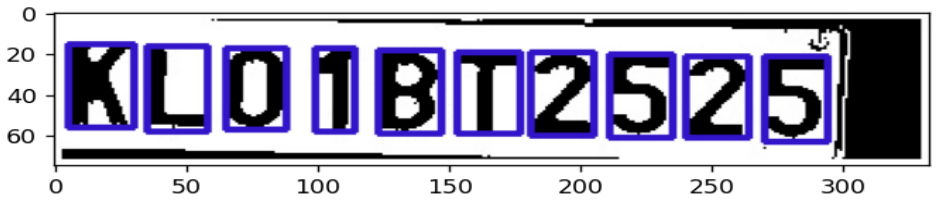


Fig. 4.5 Contours

□ Number after getting detected further converted to string

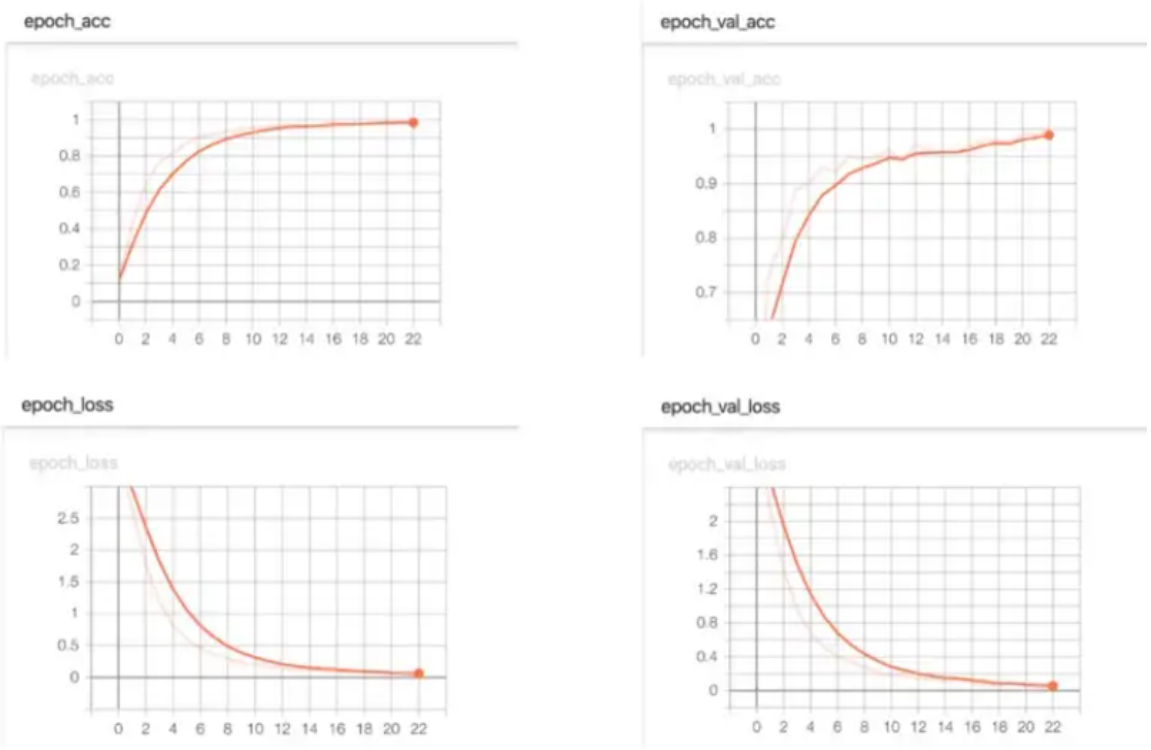
The following image shows the number of the number plate returned.

```
859/864 [============================>.] - ETA: 0s - loss: 0.1297 - custom_f1score: 0.9511
861/864 [============================>.] - ETA: 0s - loss: 0.1294 - custom_f1score: 0.9512
863/864 [============================>.] - ETA: 0s - loss: 0.1291 - custom_f1score: 0.9513
864/864 [==============================] - 49s 56ms/step - loss: 0.1290 - custom_f1score: 0.9514 - val_loss: 0.0821 - val_custom_f1score: 0.9732
XL01BT2525

Process finished with exit code 0
```

Fig. 4.6 Number plate returned after running epochs

The following graph shows after training for 23 epochs, the model achieved an accuracy of 97%.



Graph. 4.1 Training for 23 epochs

- F1 score measured is 97.32% which is the accuracy metric computes how many times a model made a correct prediction across the entire dataset.

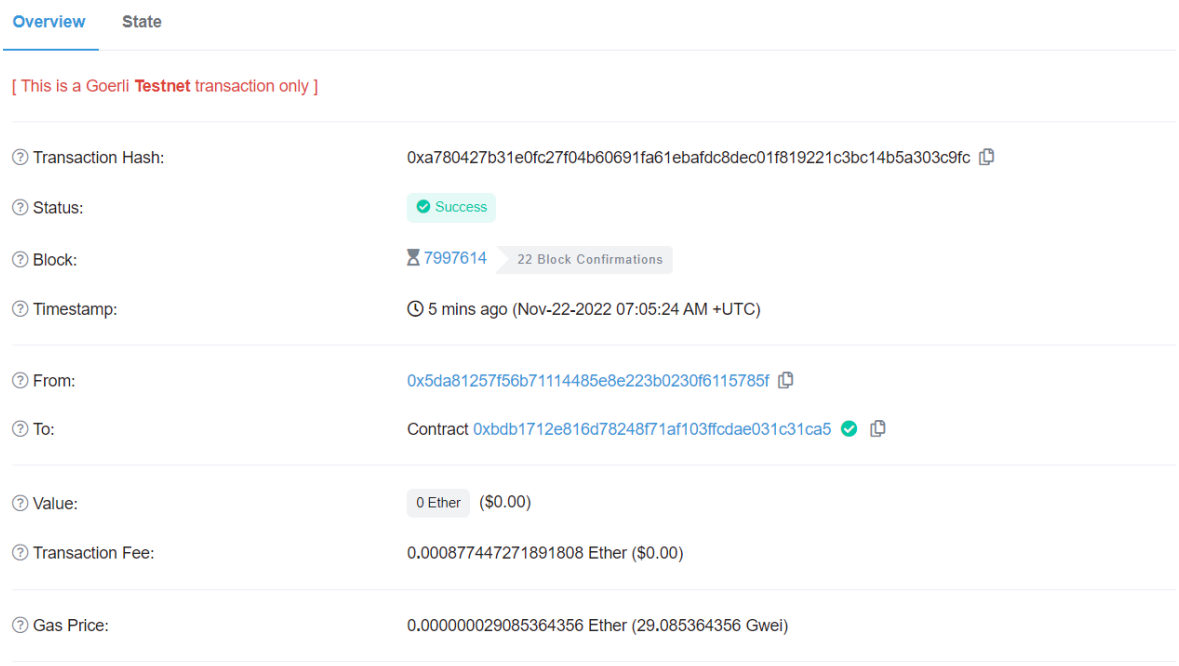


Fig. 4.7 Storage

Chapter-5: CONCLUSION

5.1 Conclusion

It is apparent that precise detection of number plates requires a comprehensive system with several phases, and that 100% overall accuracy is currently not attainable because each phase depends on the one before it. The effectiveness of number plate detection is impacted by various aspects such as various lighting conditions, vehicle shadows, non-uniform size of license plate characters, and various font and backdrop colors.

In this project, we are able to create a system that can identify an offending vehicle's license plate from photographs taken by street webcams. Then, this system uses blockchain technology to store the information from the license plate.

5.2 Application of the Project

By doing this, we can guarantee that any car that disobeys traffic laws will be stopped and given the appropriate punishment. As a result, we have a system that makes it more efficient to keep a careful eye on the roadways while using less labor.

Applications for blockchain go well beyond cryptocurrencies like bitcoin. It is a milestone in the process of integrating blockchain technology into traffic signal control systems from cryptocurrency systems. The technology is having an impact on a wide range of businesses, changing everything from how contracts are upheld to how efficiently the government is managed. In addition to saving businesses time and money, it can enhance transparency and fairness. We are guaranteed data security and outside evasion of data because we are using blockchain for storing purposes.

5.3 Future Work

For the time being, we are conducting actions and extracting photos. Future work will involve extracting videos from currently-running live traffic and using them. The YOLO framework will be used for this task. YOLO is a technique that uses neural networks to detect objects in real time. The efficiency and speed of this algorithm account for its popularity. To detect people, animals, parking meters, and traffic signals, it has been employed in a variety of applications. Additionally, the government can utilize this technology to identify the individual and the vehicle that committed a crime at a particular time and got away with it.

References

[1] [Md. Atikuzzaman, et al. 2019] Vehicle Number plate detection and categorization using CNNs. 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), 2019, pp. 1-5, doi: 10.1109/STI47673.2019.9068049.

[2] [N. Palanivel, et al. 2021] Automatic Number plate detection in vehicles using faster R-CNN. 2020 International Conference on System, Computation, Automation and Networking (ICSCAN), 2020, pp. 1-6, doi: 10.1109/ICSCAN49426.2020.9262400.

[3] [Wanxin Li, et al. 2019] A Blockchain-Based Architecture for Traffic Signal Control Systems. EEE
Communications Letters, vol. 23, no. 2, pp. 246-249, 2019.

[4] [Y. Yuan, et al. 2016] Towards blockchain-based intelligent transportation systems. 2019 IEEE International Congress on Internet of Things (ICIOT). 978-1-7281-2714-9/19/$31.00 ©2019 IEEE DOI 10.1109/ICIOT.2019.00018

[5] [C. Patel, et al.] Automatic Number Plate Recognition System (ANPR): A Survey. 2013/1/1. Journal - International Journal of Computer Applications. May 2013 International Journal of Computer Applications 69(9):21-33
DOI:10.5120/11871-7665