JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

REAL-TIME MOVING VEHICLE REGISTRATION PLATE DETECTION

[1]Rajeshri Vaidya, [2]Vaishnavi Bisen, [3]Manjusha Bansod, [4]Ganesh Masurkar, [5]Lokesh Telange, [6]Piyush Shelke

[1]Assistant Professor, [2,3,4,5,6]Final Year Bachelor Of Engineering Student
[1]Computer Science And Engineering,
[1]Sipna College Of Engineering And Technology, Amravati, Maharashtra, India

Abstract : The rapid growth in urbanization and vehicular population has brought about an increased need for efficient traffic management and enforcement systems. In this context, the development of automated systems for vehicle registration plate detection and electronic challan (e-challan) issuance has gained significant attention. This abstract presents an overview of a moving vehicle registration plate detection and e-challan system designed to enhance the effectiveness of traffic regulation and enforcement processes.

The proposed system utilizes computer vision techniques and machine learning algorithms to automatically detect and recognize vehicle registration plates in real-time. The detection process involves capturing video frames from surveillance cameras installed at strategic locations on road networks. These frames are then analyzed using image processing techniques to identify and extract the registration plates. The extracted plate images are subsequently processed by optical character recognition (OCR) algorithms to retrieve the alphanumeric characters.

IndexTerms **- Image Processing, Character Segmentation, Optical Character Recognition (OCR).**

I. INTRODUCTION

With the exponential increase in vehicular traffic and the growing need for effective traffic management, there is a pressing demand for advanced technologies that can streamline the process of vehicle registration plate detection and enforcement. Traditional manual methods of monitoring and issuing penalties for traffic violations are often time-consuming, error-prone, and inefficient. To address these challenges, the development of automated systems for moving vehicle registration plate detection and e-challan issuance has gained considerable attention.

The moving vehicle registration plate detection and e-challan system leverages computer vision techniques, image processing algorithms, and machine learning models to automatically detect and recognize vehicle registration plates in real-time. By employing strategically placed surveillance cameras, this system captures video frames of moving vehicles, analyzes them, and extracts the registration plate information. The extracted plate data is then processed using optical character recognition (OCR) algorithms, enabling the retrieval of alphanumeric characters present on the plate.

Once the registration plate information is successfully obtained, the system compares it with a pre-existing database of registered vehicles. This comparison allows for the identification of any violations or irregularities associated with the detected vehicle, such as expired registrations, unauthorized parking, or other traffic offenses. In case of a violation, the system generates an e-challan, which serves as an electronic ticket documenting the offense committed, the time and location of the incident, and the necessary details of the vehicle owner.

The e-challan system replaces the conventional paper-based process by issuing penalties electronically to the vehicle owners. This digital approach eliminates the need for physical paperwork and enables a seamless and efficient enforcement process. It also provides an electronic trail of violations, making it easier for authorities to track and manage traffic offenses. Furthermore, the system can integrate with existing databases, allowing law enforcement agencies to maintain a comprehensive record of traffic violations for better monitoring and analysis.

The implementation of a moving vehicle registration plate detection and e-challan system offers numerous advantages. Firstly, it significantly reduces the reliance on manual methods for registration plate identification, minimizing human errors and improving accuracy in enforcing traffic regulations. Secondly, the real-time nature of the system enables immediate identification of violators, enabling timely enforcement actions and contributing to improved traffic flow and safety. Additionally, the electronic issuance of e-challans reduces administrative burdens, eliminates paperwork, and promotes a more sustainable and eco-friendly approach to traffic management[1].

II. LITERATURE REVIEW

There are several studies and research papers on license plate detection and E-challan systems, highlighting their benefits and limitations. Some of the notable literature reviews are:

1. "License Plate Detection and Recognition Using Deep Learning" by Yash Shah and Gaurav Patel: This paper discusses the use of deep learning algorithms for license plate detection and recognition. The authors compare different deep learning architectures and evaluate their performance on real-world datasets.
2. "Development and Implementation of Electronic Challan System for Traffic Violation Detection" by Rajesh Kumar and Rakesh Kumar: This paper presents the implementation of an E-challan system in India and discusses its benefits and challenges. The authors also compare the E-challan system with traditional manual methods and evaluate its effectiveness.
3. "A Survey on License Plate Recognition Systems" by Adithya M. and S. Sowmya: This paper provides a comprehensive review of license plate recognition systems, including their architectures, algorithms, and applications. The authors discuss the different approaches used for license plate detection, segmentation, and recognition and highlight their advantages and limitations.
4. "Automated Traffic Law Enforcement: A Review of Technical and Social Issues" by Hina Tabassum and Shehzad Khalid: This paper provides a critical review of automated traffic law enforcement systems, including license plate detection and E-challan systems. The authors discuss the technical and social issues associated with these systems, including privacy concerns, accuracy, and reliability. Overall, the literature review suggests that license plate detection and E-challan systems have several benefits over traditional manual methods, including increased efficiency, accuracy, and convenience. However, these systems also raise privacy concerns and require an adequate infrastructure to function effectively. Therefore, proper measures should be taken to address the technical and social issues associated with these systems to ensure their effectiveness and reliability.Searching for license plate recognition is still a challenge. It involves three major steps. They specify number pad space, character segmentation, and character recognition. Each step suggested different ways to improve efficiency.

One of these methods used the adaptive threshold to highlight the characters and suppress the background. In order to remove unwanted image spaces, a component algorithm is first applied to the converted binary image from the original panel. A special algorithm called Image Scissoring is used to divide the Optical Character Recognition engine called tesseract, which returns ASCII to the license number. The entire system has been implemented using open CV. Another method is to deploy the forward background feed method for character classification. The neural network is developed by using the backward-propagation algorithm. Normalization, scale and edge detection are included in the steps of the pre-processing. The horizontal and vertical graph and component survey are able to address the problem of character fragmentation.

Another way in which character areas are selected is through binarization, connected component analysis. The Point Analysis method removes unwanted points and combines split points and split points. This unit achieves a 97.2% accuracy rate in character segmentation. The reliability of the recognition was 90.9%. Offers an approach that relies on effective morphological operation and the detection method of Sobel Edge. This approach is simplified to divide all letters and numbers used in the number pad using the surround box method. After the template is fragmented, the matching policy is used to recognize numbers and characters. This whole system was implemented using MATLAB.

Provides an overview of the analysis of related components and processes, such as aspect ratio analysis and pixel count analysis. In the author studies a comparison of four algorithms that are sequentially using statistical properties, the Hough Transform and Contour algorithm, the medium transformation approach and morphological processes and their results.

The handwritten text is fragmented by the watershed algorithm. Noise removal, slope correction, budgeting and normalization were eliminated in pre-treatment. After
fragmentation the process of extracting a segmented image is done by a reverse integer to convert the wavelet integer.The classification is then sorted by neuroscience.

MATLAB VS OPENCV At present, open CV is a great dealing with the open-source library for computer vision and has a large community of users. Open CV has much more functionality to see the computer than MATLAB. Many of their functions are performed on the GPU. The library is updated continuously (a new version is released every 3 to 4 months). In general, the open CV C ++ program can be executed with a high speed than the MATLAB code.

OpenCV has more functions to see the computer than MATLAB. Many of their functions are performed on the GPU. The C ++ Open CV code is usually run faster than the MATLAB code, but compared to open CV C ++, open CV is much better than C ++. Python is better and easier than other programming languages like C ++ in seeing the computer, we encounter similar options. What a tool you should learn Engineer / Programmer Computer vision – Open CV using C ++, or Open CV using Python, or MATLAB, as at present we have some options to choose from. In the past there were no

good libraries to see the computer. We identified these studies by means of relevant books that were available and began coding the special library of special algorithms for computer vision Like MATLAB, Open CV is also made for image processing and used as an alternative tool and much faster than other simulations. Each function is designed in Open CV, the function structure and data using the image processing coding software. On the other hand, we get nearly everything in the world in the form of toolboxes on MATLAB.

Although MATLAB is a relatively simple language, this high-level programming language has become slower in some cases. In such cases, open CV works better and produces accurate results. Similarly, it can be very simple to handle some code to model the idea of processing your images. One of the outstanding contributions of the Open-Source community in the scientific world is Python.

The main concepts for building an algorithm like the one described in this paper comes from the application of object detection alongside with digital image processing. The object detection is characterized as a source of identification of instances or classes for an object inside an image and the main goal is to find all these classes and 2 instances.
This may be used to innumerous cases, such as a car detection or any other vehicle. The digital image processing is a method where operations are performed on a digital image to extract information or get a new image with enhanced features.

Also, there are innumerous applications, such as finding the boundaries between types of tissue in medicine or detecting the contour of characters and numbers, in the case of an OCR system. The combination of these techniques makes almost all kinds of pattern recognition possible inside an image. In that way, they may be responsible to guide an autonomous quadrotor during an active car tracking or any kind of tracking / detection mission. In the field of object tracking, there are

innumerous papers and researches that have inspired the development of the algorithm presented in this paper. Barták and Vyškovský, presented a Track Learn and Detect - TLD solution for object tracking and following implemented on a Parrot AR Drone, that uses pixels comparison for each frame. Patil implemented a version of the CamShift algorithm where it uses angles formed in the camera as the source of the centroid coordinates of an object, alongside with a NavStik autopilot board. At the same time, among the OCR researches,Tavares, Causin and Gonzaga presented an algorithm for Brazilian car plates recognition using Google's Tesseract platform. Alongside with them, Qadri and Asif used the Automatic Number Plate Recognition – ANPR to extract information from car plates and turn them into characters and numbers. Without the ideas presented in those researches, it wouldn't be possible to create the algorithm present in this paper.

III. OBJECTIVES

1. **Improve Road Safety:** The primary objective of the system is to promote safe driving and reduce road accidents. By detecting and penalizing traffic violations, the system aims to discourage reckless driving behavior and encourage compliance with traffic rules.
2. **Enhance Traffic Enforcement:** The system aims to improve the effectiveness and efficiency of traffic enforcement by automating the process of issuing fines for traffic violations. This eliminates the need for manual enforcement, reducing the chances of corruption and inefficiencies.
3. **Reduce Traffic Violations:** The system aims to reduce the incidence of traffic violations such as overspeeding, jumping red lights, and driving without a helmet or seat belt. By detecting and penalizing violators, the system aims to deter them from repeating the offense and promote safe driving behavior.
4. **Increase Transparency and Accountability:** The system promotes transparency and accountability in traffic enforcement by generating an auditable record of traffic violations and fines issued. This improves trust and confidence in law enforcement agencies and reduces the chances of corruption and malpractice.
5. **Enhance Data Collection and Analysis:** The system generates real-time reports on traffic violations and their locations, which can be used to identify high-risk areas and take appropriate measures to improve road safety. The data collected can also be used for statistical analysis and research on traffic patterns and trends.

Overall, the objectives of a moving vehicle registration plate detection and e-challan system are to improve road safety, enhance traffic enforcement, reduce traffic violations, increase transparency and accountability, and enhance data collection and analysis.

IV. SYSTEM ANALYSIS

4.1 Proposed System:

The methodology for the license plate detection and E-challan system involves the following steps:

1. **Infrastructure Setup:** The first step involves setting up the necessary infrastructure for the system, which includes installing cameras at strategic locations on roads and highways. These cameras should be positioned in such a way that they can capture clear images of passing vehicles.

2. **Image Processing:** Once the images are captured, the next step involves processing the images to detect the license plate number. This is achieved through the use of image processing algorithms that analyse the images and identify the license plate numbers.

3. **License Plate Verification:** Once the license plate number is detected, the system cross-checks it with the database of registered vehicles to determine if the vehicle is authorized to be on the road or if it has any pending traffic violations.

4. **E-Challan Generation:** If the vehicle is found to have violated any traffic rules, an electronic challan is generated and sent to the registered address of the vehicle owner. The electronic challan includes details of the violation, the amount of the fine, and the due date for payment.

5. **Payment and Penalties:** The system provides a paperless and cashless system for traffic violation fines, making it more convenient for both the traffic police and the vehicle owners. The vehicle owner can pay the fine through various electronic payment methods, including online banking, credit card, or debit card.

6. **Monitoring and Analysis:** The system continuously monitors the traffic flow and records the number of violations detected. This data can be analysed to identify the areas with the
highest number of violations, which could help in improving road safety by taking necessary measures to reduce violations in those areas. The methodology for the license plate detection and E- challan system involves a combination of hardware and software technologies, including cameras, image processing algorithms, and databases. The system requires an adequate infrastructure to function effectively and efficiently. Proper measures should also be taken to ensure the privacy and security of the collected data[2,3].

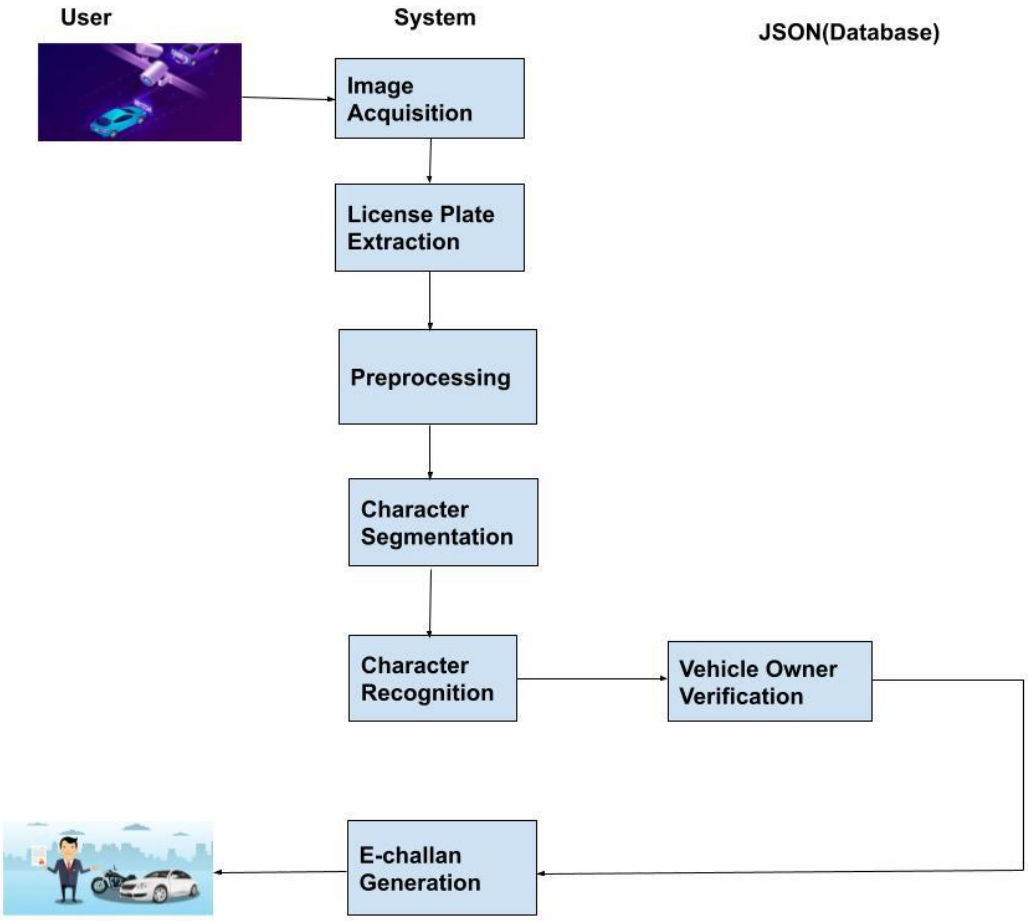


Fig 4.1: Flow of Proposed Work

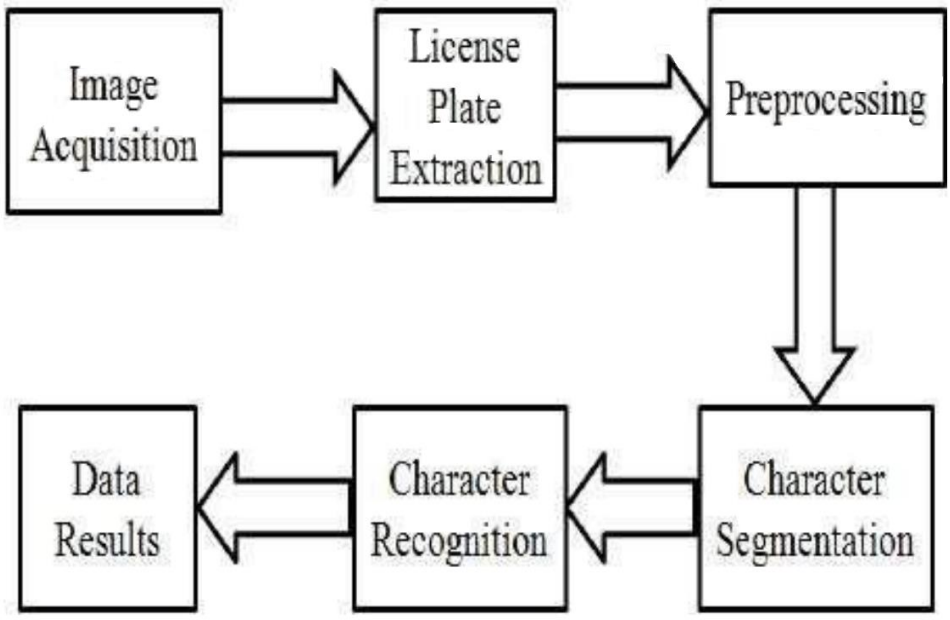


Fig 4.2: Working of Number Plate Detection

In the proposed work there are 2 modules i.e., user and controller/ admin. User can view the challan against its vehicle and pay the amount through QR code, which is the most convenient way now a days to do online transaction. And on the other hand, admin or controller can add new challan or assign challan, according to the number capture from the video.

To view the number plate from the video, firstly the images are extracted from the video, then from that images the number plate get extracted for pre-processing of data, once this done the character segmentation is done for Recognition of character and through this process we can get the clear image of the number plate.

Optical Character Recognition is a technique that deals with recognition of optically drawn characters and is needed when information must be read by both humans and computers when another kind of input is not given. OCR is a complex problem because of the variety of languages, fonts and styles in which texts can be written, and the complex rules of languages and needs a sort of steps to achieve reasonable results. Inside the major steps for an OCR algorithm, we have image acquisition, where the data is feed; pre-processing, where the quality of the image is improved, and character segmentation, where characters are separated, where they compose the first stage of the algorithm.

Steps of OCR. (a) Image acquisition. (b) Pre-processing. (c) Character segmentation. (d) Feature extraction. (e) Classification and post-processing. The second stage are composed by feature extraction, where each character segment is processed to obtain unique features; classification, where the mapping between segmented images to classes are made, and post-processing, that is the refine stage to guarantee the right result.

4.2 Technologies Used

For the proposed study We have used many python in-built libraries and modules such as OpenCV, NumPy, Keras ocr, HTML CSS, Flask and tKinter.

4.2.1 OpenCV-Python:-

OpenCV-Python is a library of Python bindings designed to solve computer vision problems. OpenCV-Python makes use of NumPy, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from NumPy arrays. This also makes it easier to integrate with other libraries that use NumPy such as SciPy and Matplotlib.

In OpenCV, the CV is an abbreviation form of a computer vision, which is defined as a field of study that helps computers to understand the content of the digital images such as photographs and videos.

The purpose of computer vision is to understand the content of the images. It extracts the description from the pictures, which may be an object, a text description, and three-dimension model, and so on. For example, cars can be facilitated with computer vision, which will be able to identify and different objects around the road, such as traffic lights, pedestrians, traffic signs, and so on, and acts accordingly.

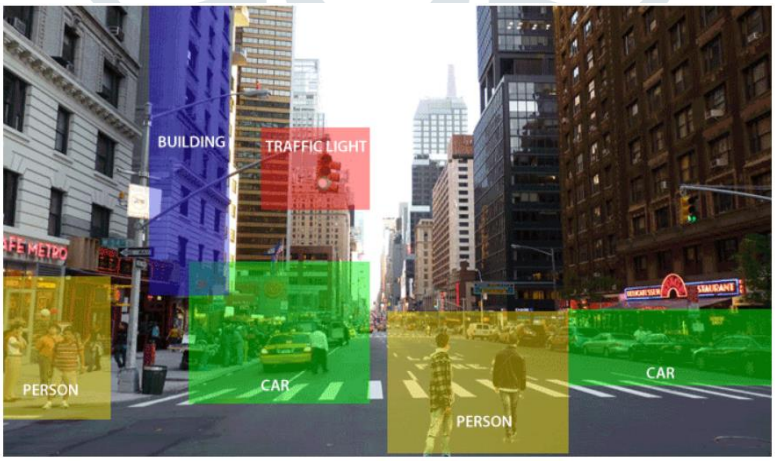


Fig 4.2.1.1: OpenCV example of identification model

Computer vision allows the computer to perform the same kind of tasks as humans with the same efficiency. There are a two main task which are defined below:

i. **Object Classification -** In the object classification, we train a model on a dataset of particular objects, and the model classifies new objects as belonging to one or more of your training categories.
ii. **Object Identification -** In the object identification, our model will identify a particular instance of an object.

OpenCV installation
pip install OpenCV-python

Read & Save Images
Now for OpenCV to work on any image, it must be able to read it. Here we will see how to read a file and save it after we are done with it. Let's see how to do it: Imread function in OpenCV.

We use the imread function to read images, here is the syntax of this function

cv2.imread (path, flag)

The path parameter takes a string representing the path of the image to be read. The file should be in the working directory or we must give the full path to the image. The other parameter is the flag which is used to specify how our image should be read.

cv2.imread (path, flag)

The path parameter takes a string representing the path of the image to be read. The file should be in the working directory or we must give the full path to the image. The other parameter is the flag which is used to specify how our image should be read.

Inwrite function in OpenCV
We can use OpenCV's imwrite() function to save an image in a storage device and the file extension defines the image format as shown in the example below. The syntax is the following:

cv2.imwrite (filename, image)

OpenCV Video Capture

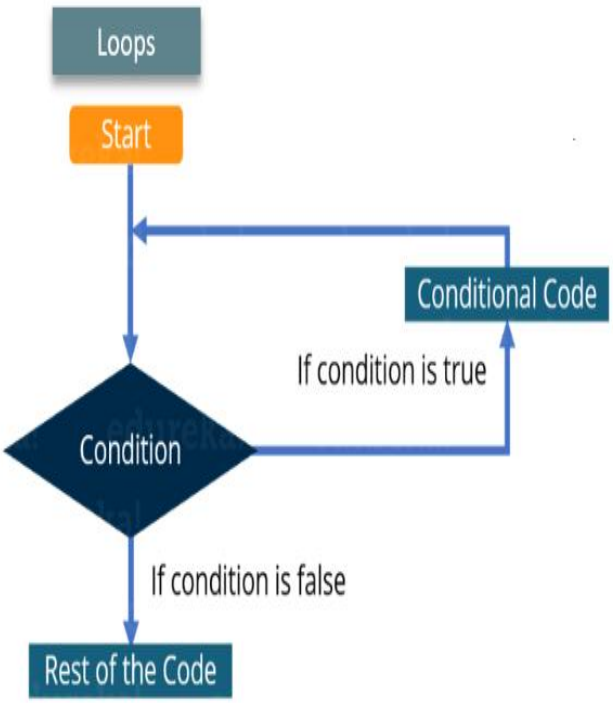


Fig 4.2.1.2: How to capture video using OpenCV

Capture Video from Camera

Often, we have to capture a live stream with a camera. Using OpenCV's very simple interface, we can easily do it. Here is a simple task to get started. In this task we will capture a video from the camera (in-built webcam of my laptop) and display it as a grayscale video.

In OpenCV we need to create a VideoCapture object to capture a video. We pass either the device index or the name of a video file as its arguments. Device index is just the number to specify the camera in case we have multiple webcams available. Normally one has only a single camera connected (as in my case), so simply pass 0.After this we start to capture each frame using a loop and process it accordingly .At the end, we just break from the loop and release the capture[6].

Saving a Video

Saving an image after processing it using OpenCV is quite simple and we saw how to do it using cv2.imwrite () function. But for a video, after processing it frame-by-frame a little more work is required to save it.
Here to save a video we create a VideoWriter object in addition to VideoCapture Object. The syntax of VideoWriter is given below:

cv2.VideoWriter(filename, fourcc, fps, frameSize, isColor)

Parameters:

i. **filename:** the output file name (eg: bday.avi).
ii. **fourcc:** specify the FourCC code. FourCC is a 4-byte code used to specify the video codec. The list of available codes can be found in fourcc.org. It is platform dependent. Following codecs works fine for me.
iii. **fps:** number of frames per second (fps)
iv. **frameSize:** size of frame.
v. **isColor:** it is a flag value. If it is True, encoders expect a color frame, otherwise it works with grayscale frames.

Face Detection Using OpenCV

Using OpenCV, complex tasks such as face detection becomes easy to implement and since pre-trained models that are capable of detecting faces, noses, and eyes are included in the OpenCV package, we don't need to train any classifier. Here is an article on Face detection using Viola-Jones algorithm that explains how we can detect faces using OpenCV.You will also go through the concept of cascading classifier in this article that is also used in our next section i.e. car detection using OpenCV[4].

4.2.2 Keras

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research.

Keras is:

- Simple -- but not simplistic. Keras reduces developer cognitive load to free you to focus on the parts of the problem that really matter.
- Flexible -- Keras adopts the principle of progressive disclosure of complexity: simple workflows should be quick and easy, while arbitrarily advanced workflows should be possible via a clear path that builds upon what you've already learned.
- Powerful -- Keras provides industry-strength performance and scalability: it is used by organizations and companies including NASA, YouTube, or Waymo.

Keras is the high-level API of TensorFlow 2: an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity.

keras-OCR

Optical character recognition or optical character reader (OCR) is the electronic or mechanical conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo (for example the text on signs and billboards in a landscape photo). Widely
used as a form of data entry from printed paper data records – whether passport documents, invoices, bank statements, computerized receipts, business cards, mail, printouts of static-data, or any suitable documentation – it is a common method of digitizing printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as cognitive computing, machine translation, (extracted) text-to-speech, key data and text mining.

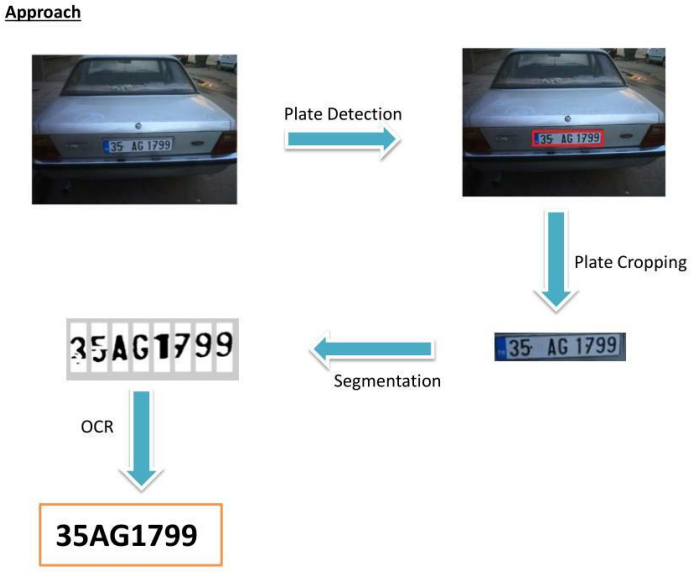


Fig 4.2.2.1 Steps of OCR- Number Plate Detection

Optical Character Recognition is a technique that deals with recognition of optically drawn characters and is needed when information must be read by both humans and computers when another kind of input is not given. OCR is a complex problem because of the variety of languages, fonts and styles in which texts can be written, and the complex rules of languages and needs a sort of steps to achieve reasonable results. Inside the major steps for an OCR algorithm, we have image acquisition, where the data is feed; pre-processing, where the quality of the image is improved, and character segmentation, where characters are separated, where they compose the first stage of the algorithm.

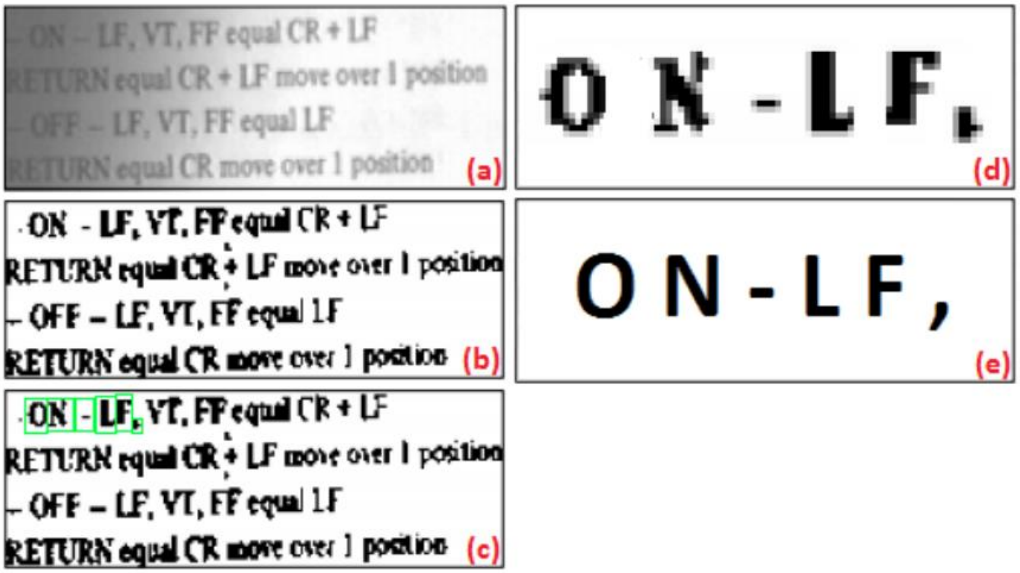


Fig 4.2.2.1: Steps of OCR. (a) Image acquisition. (b) Pre-processing. (c) Character segmentation. (d) Feature extraction. (e) Classification and post-processing[10].

The second stage are composed by feature extraction, where each character segment is processed to obtain unique features; classification, where the mapping between segmented images to classes are made, and post-processing, that is the refine stage to guarantee the right result[10].

5.2.3 HTML

HTML stands for hypertext markup language. It's made of keywords and commands that web designers use for creating websites.

Hypertext is text with links that readers can simply click on to go to another page or another part of the page. Meanwhile, markup language uses tags or plain text with special markings to define the sections of a page, such as headers and footers, and other elements, including tables and images.

HTML is considered one of the three essential tools in webpage creation: HTML provides the structure or the way text, pictures, and so on will appear on the website. CSS (cascading style sheets) sets the visual properties of these elements, such

as colors, format, and layout. Meanwhile, Javascript makes these elements behave in certain ways depending on a user's action. For example, the font size of text can increase when users hover their mouse or click a button on a page.

HTML is the default language of websites and web-based documents. It helps a browser understand the structure and style of a document or files for viewing over the internet. It allows your web pages to host audio, videos, spreadsheets, and other applications. It also facilitates navigation within web pages or between websites through hypertext.
Moreover, website makers can use HTML to design forms for ordering products, making reservations, or searching for information. HTML is, therefore, the basic building block for building your brand and running an e-commerce site or an online subscription-based business.

5.2.4 CSS

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL. CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.

5.2.5 Flask Framework

The framework is the basis upon which software programs are built. It serves as a foundation for software developers, allowing them to create a variety of applications for certain platforms. It is a set of functions and predefined classes used to connect with the system software and handle inputs and outputs.

It simplifies the life of a developer while giving them the ability to use certain extensions and makes the online applications scalable and maintainable.

Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2. Advantages of using Flask framework are:

- There is a built-in development server and a fast debugger provided.
- Lightweight
- Secure cookies are supported.
- Templating using Jinja2.
- Request dispatching using REST.
- Support for unit testing is built-in.

5.2.6 Tkinter

Tkinter is the Python port for **Tcl-Tk GUI toolkit** developed by Fredrik Lundh. This module is bundled with standard distributions of Python for all platforms. Tkinter is the Python interface to Tk, which is the GUI toolkit for Tcl/Tk. Tcl (pronounced as tickle) is a scripting language often used in testing, prototyping, and GUI development. Tk is an open-source, cross-platform widget toolkit used by many different programming languages to build GUI programs. Python implements the Tkinter as a module. Tkinter is a wrapper of C extensions that use Tcl/Tk libraries. Tkinter allows you to develop desktop applications. It's a very good tool for GUI programming in Python.
Tkinter is a good choice because of the following reasons:

- Easy to learn.
- Use very little code to make a functional desktop application.
- Layered design.
- Portable across all operating systems including Windows, macOS, and Linux.
- Pre-installed with the standard Python library.

Developing desktop-based applications with python Tkinter is not a complex task. An empty Tkinter top-level window can be created by using the following steps.

i. import the Tkinter module.
ii. Create the main application window.
iii. Add the widgets like labels, buttons, frames, etc. to the window.
iv. Call the main event loop so that the actions can take place on the user's computer screen.

Tcl/Tk is not a single library but rather consists of a few distinct modules, each with separate functionality and its own official documentation. Python's binary releases also ship an add-on module together with it.

Tcl

Tcl is a dynamic interpreted programming language, just like Python. Though it can be used on its own as a general-purpose programming language, it is most commonly embedded into C applications as a scripting engine or an interface to the Tk toolkit. The Tcl library has a C interface to create and manage one or more instances of a Tcl interpreter, run Tcl commands and scripts in those instances, and add custom commands implemented in either Tcl or C. Each interpreter has an event queue, and there are facilities to send events to it and process them. Unlike Python, Tcl's execution model is designed around cooperative multitasking, and Tkinter bridges this difference.

Tk

Tk is a Tcl package implemented in C that adds custom commands to create and manipulate GUI widgets. Each Tk object embeds its own Tcl interpreter instance with Tk loaded into it. Tk's widgets are very customizable, though at the cost of a dated appearance. Tk uses Tcl's event queue to generate and process GUI events.

Ttk

Themed Tk (Ttk) is a newer family of Tk widgets that provide a much better appearance on different platforms than many of the classic Tk widgets. Ttk is distributed as part of Tk, starting with Tk version 8.5. Python bindings are provided in a separate module, tkinter.ttk. Internally, Tk and Ttk use facilities of the underlying operating system, i.e., Xlib on Unix/X11, Cocoa on macOS, GDI on Windows.When your Python application uses a class in Tkinter, e.g., to create a widget, the tkinter module first assembles a Tcl/Tk command string. It passes that Tcl command string to an internal _tkinter binary module, which then calls the Tcl interpreter to evaluate it. The Tcl interpreter will then call into the Tk and/or Ttk packages, which will in turn make calls to Xlib, Cocoa, or GDI[4].

V. IMPLEMENTATION

The implementation of a moving vehicle registration plate detection and e-challan system involves several steps, which can be detailed as follows:

1. **Hardware and software requirements:** The first step is to identify the hardware and software requirements of the system. This includes cameras and software for vehicle registration plate detection, a database management system for storing vehicle and owner information, software for generating electronic fines, and a user interface for law enforcement agencies.
2. **Design of the system architecture:** The next step is to design the system architecture, which includes the different modules and their interactions. This involves creating a flowchart and designing the database schema.
3. **Development of the system:** Once the system architecture is designed, the next step is to develop the system. This includes coding the different modules, integrating them, and testing the system for functionality and performance.
4. **Installation and testing of the hardware:** The hardware components, such as the cameras, need to be installed and tested to ensure that they are working correctly. This involves testing for image quality, angle, and lighting conditions.
5. **Integration with other systems:** The moving vehicle registration plate detection and e-challan system needs to be integrated with other traffic management systems, such as toll booths, traffic signals, and surveillance cameras. This involves developing APIs for data exchange and testing the integration.
6. **Deployment and training:** The system needs to be deployed in the field, and law enforcement agencies need to be trained in using the system. This involves creating user manuals and training materials.
7. **Data analysis and evaluation:** The system needs to be evaluated for its effectiveness in reducing traffic violations and promoting road safety. This involves analyzing the data collected and comparing it with pre-implementation data.

Overall, the implementation of a moving vehicle registration plate detection and e-challan system involves several steps, starting from identifying the hardware and software requirements to evaluating the system's effectiveness[5].

This process can be used for a thesis, with a focus on a specific aspect of the system, such as the effectiveness of the system in reducing traffic violations or the integration of the system with other traffic management systems.

5.1 Code Implemetation

5.1.1 sever.py

```
def check(vehicleNo, score):
   if vehicleNo.startswith('MH') and len(vehicleNo) >= 8:
      # check if vehicleNo is valid

      myFile = open('data.json')
      data = json.load(myFile)
      myFile.close()

      if not any(d['vehicleNo'] == vehicleNo for d in data):
         print("New vehicleNo Adding to file", vehicleNo, score)
         now = datetime.now().strftime("%d/%m/%Y %H:%M:%S")
         charge = getRandomCharge()
         data.append({'vehicleNo': vehicleNo, 'time': now,
                  'charge': charge, 'paid': False})

         with open('data.json', 'w') as f:
            json.dump(data, f, indent=4)

         print("Sending Email to user..")
```

```
            mailgun.mail("New challan generated", "Hello, Your vehicleNo " + vehicleNo +
                         " has been detected by our system at " + now + ". Please pay the challan of Rs. " + str(charge) + " to avoid any
legal action.")

        else:
            print("vehicleNo Already in file")


def getSwitchValue():
    myFile = open('switch.txt')
    data = myFile.read()
    myFile.close()
    return int(data)


def readVideo(filepath):

    cam = cv2.VideoCapture(filepath)
    print(filepath)
    cv2.namedWindow("video")


    while cam.isOpened():

        ret, frame = cam.read()
        if not ret:
            continue

        frame = cv2.resize(frame, (350, 300))

        text_ = reader.readtext(frame)

        for t_, t in enumerate(text_):
            # print(t)

            bbox, text, score = t

            if score > threshold:

                try:
                    cv2.putText(
                        frame, text, bbox[0], cv2.FONT_HERSHEY_COMPLEX, 0.65, (255, 0, 0), 2)
                except:
                    pass

                if getSwitchValue() == 1:
                    check(text, score)
                else:
                    print("Switch is off")

        cv2.imshow("video", frame)

        k = cv2.waitKey(1)
        if k == ord('q'):
            print("q pressed, quitting...")
            break

    cam.release()

    cv2.destroyAllWindows()

@app.route('/upload_video.html')
def uploadvideo():
    return render_template('upload_video.html')

@app.route('/<path:path>')
def serve_static(path):
    if os.path.isdir(os.path.join(static_path, path)):
```

```
        path = os.path.join(path, 'index.html')
    return send_from_directory(static_path, path)


@app.route('/getItems')
def getItemsJson():
    return jsonify(getItemsList())


@app.route('/getSwitch')
def getSwitch():
    with open('switch.txt', 'r') as f:
        return jsonify({'switch': int(f.read())})


@app.route('/setSwitch', methods=['POST'])
def setSwitch():
    reqjson = request.get_json()

    val = reqjson['switch']


    with open('switch.txt', 'w') as f:
        f.write(str(val))

    return jsonify({'msg': 'ok'})


@app.route('/deleteItem', methods=['POST'])
def deleteFromJson():
    reqjson = request.get_json()
    vehicleNo = reqjson['vehicleNo']
    deleteItem(vehicleNo)
    return jsonify({'msg': 'ok'})


@app.route('/pay-challan', methods=['POST'])
def payChallan():
    reqjson = request.get_json()
    vehicleNo = reqjson['vehicleNo']
    payTheChallan(vehicleNo)
    return jsonify({'msg': 'ok'})


@app.route('/predict-video', methods=['POST'])
def predictVideo():
    myfile = request.files['vidfile']
    filename = myfile.filename
    filepath = os.path.join(app.config['UPLOAD_FOLDER'], filename)
    myfile.save(filepath)
    readVideo(filepath)
    os.remove(filepath)


    return jsonify({'msg': 'ok'})

if __name__ == '__main__':
    app.run(debug=True)
```

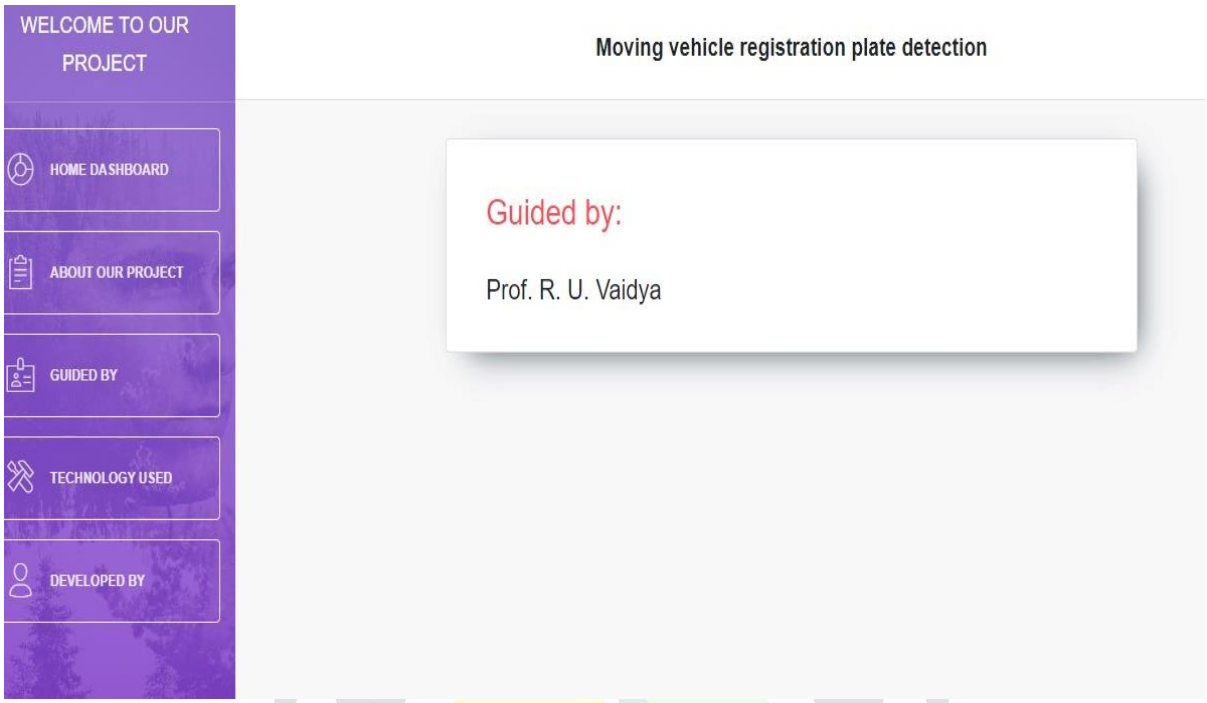
5.1.2 camera.py

```
def getRandomCharge():
    charges = [100, 200, 300, 400, 500]
    return charges[random.randint(0, len(charges) - 1)]


def check(vehicleNo, score):
    if vehicleNo.startswith('MH') and len(vehicleNo) >= 8:
```

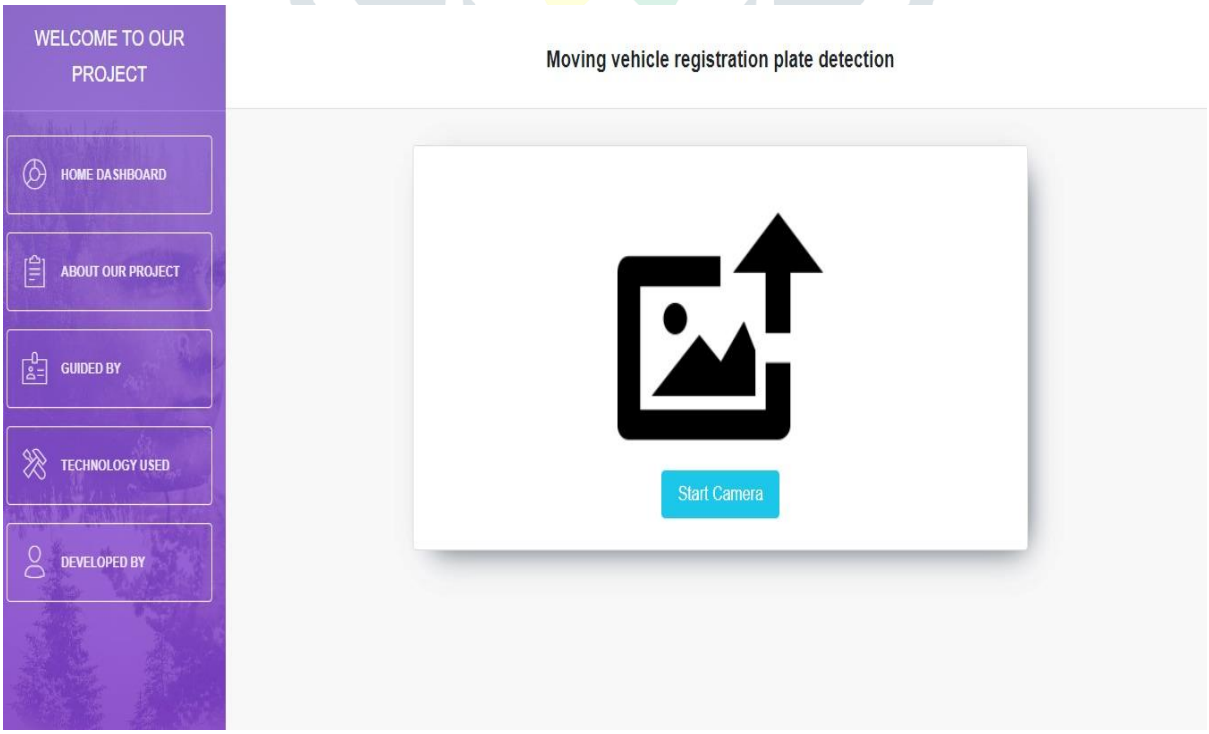
```
        # check if vehicleNo is valid

        myFile = open('data.json')
        data = json.load(myFile)
        myFile.close()

        if not any(d['vehicleNo'] == vehicleNo for d in data):
            print("New vehicleNo Adding to file", vehicleNo, score)
            now = datetime.now().strftime("%d/%m/%Y %H:%M:%S")
            charge = getRandomCharge()
            data.append({'vehicleNo': vehicleNo, 'time': now,
                         'charge': charge, 'paid': False})

            with open('data.json', 'w') as f:
                json.dump(data, f, indent=4)


            print("Sending Email to user..")
            mailgun.mail("New challan generated", "Hello, Your vehicleNo " + vehicleNo +
                         " has been detected by our system at " + now +". Please pay the challan of Rs. " + str(charge) + " to avoid any
legal action.")

        else:
            print("vehicleNo Already in file")


def getSwitchValue():
    myFile = open('switch.txt')
    data = myFile.read()
    myFile.close()
    return int(data)


def start_cam():
    while True:
        ret, frame = cam.read()

        if not ret:
            continue

        frame = cv2.resize(frame, (350, 300))

        text_ = reader.readtext(frame)

        for t_, t in enumerate(text_):
            # print(t)

            bbox, text, score = t

            if score > threshold:

                try:
                    cv2.putText(
                        frame, text, bbox[0], cv2.FONT_HERSHEY_COMPLEX, 0.65, (255, 0, 0), 2)
                except:
                    pass

                if getSwitchValue() == 1:
                    check(text, score)
                else:
                    print("Switch is off")

        cv2.imshow("cam", frame)

        k = cv2.waitKey(1)
        if k == ord('q'):
            print("q pressed, quitting...")
            break
```

```
    cam.release()

    cv2.destroyAllWindows()

if __name__ == '__main__':
    start_cam()
```

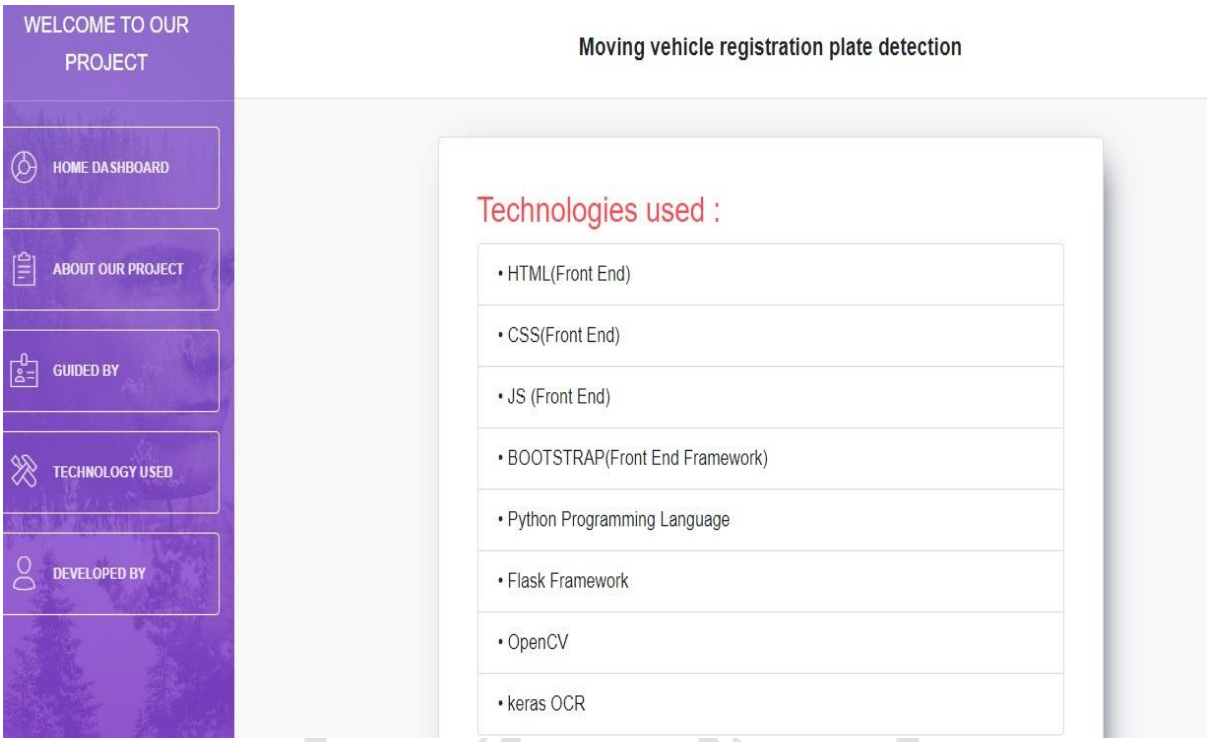
VI. RESULT



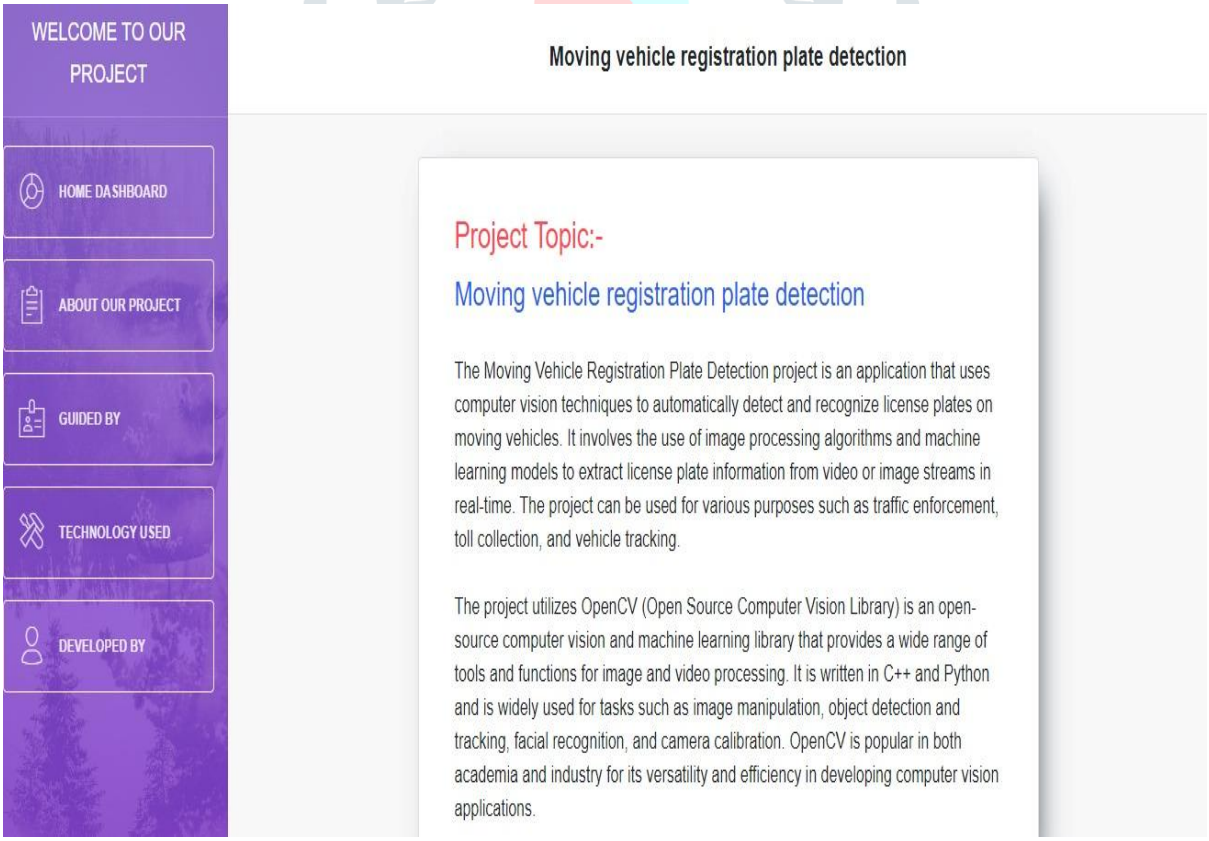
Screenshot 6.2.1: Home Page



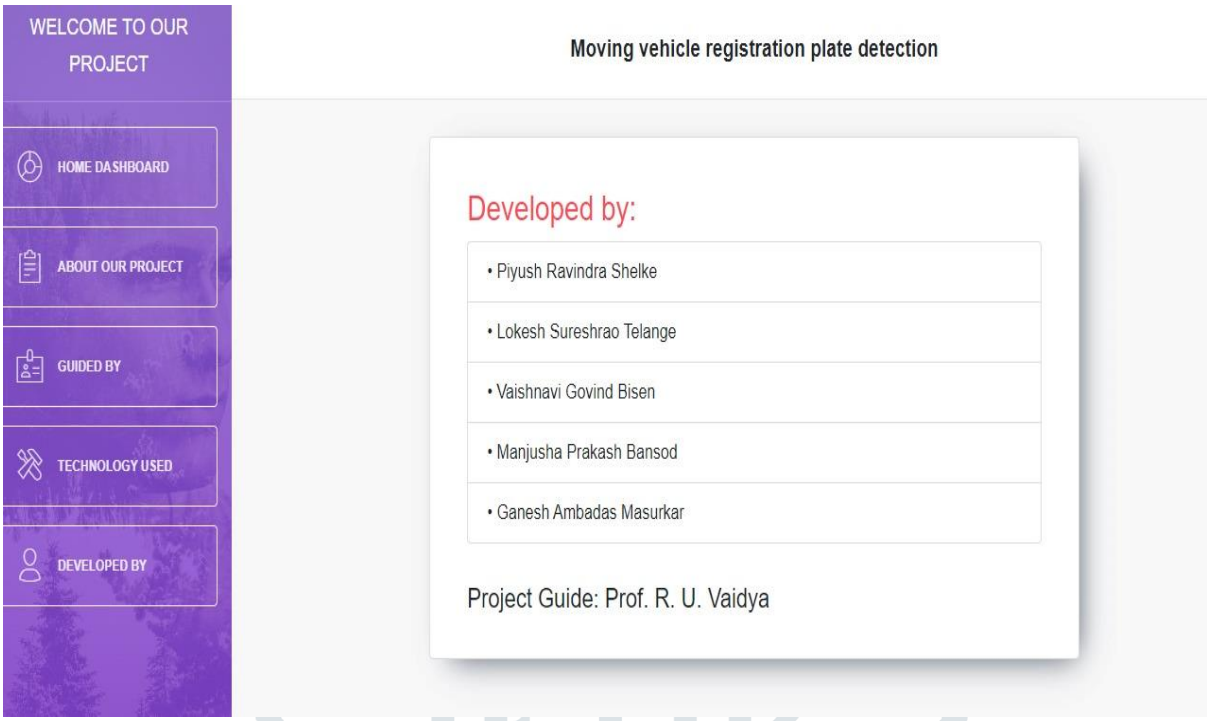
Screenshot 6.2.2: Process of Recording a Video



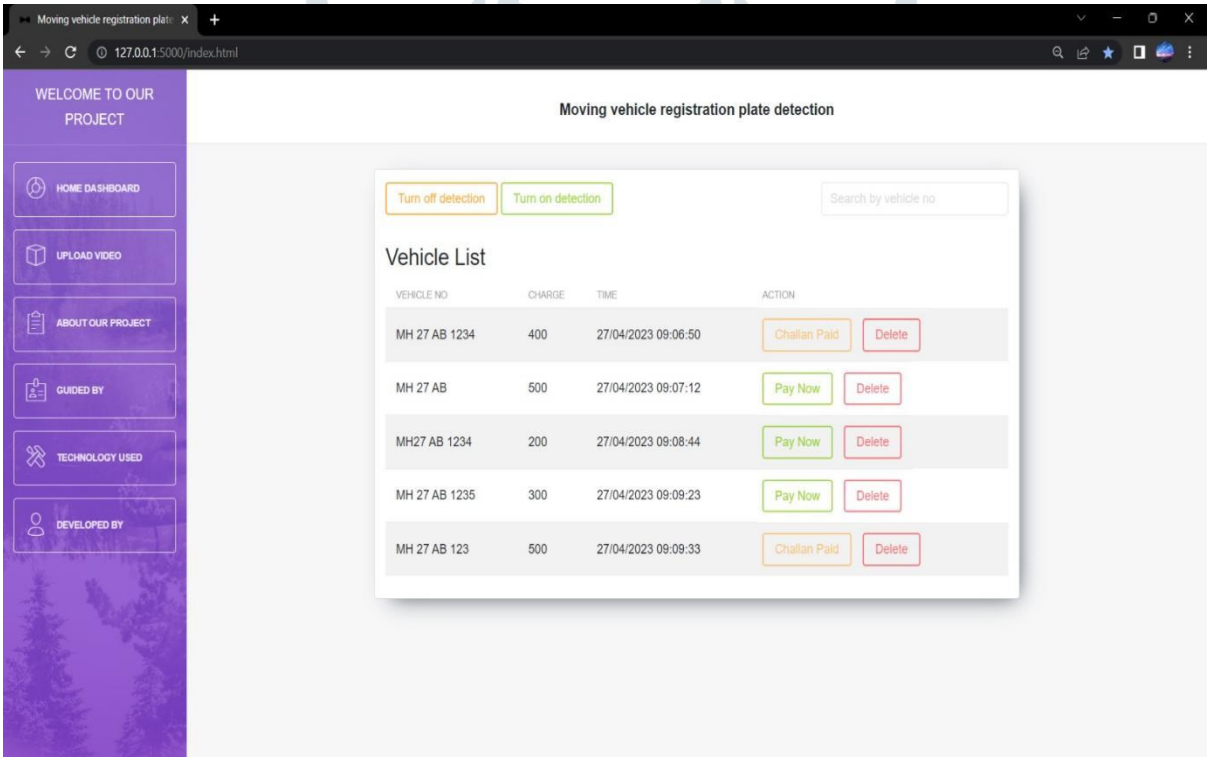
Screenshot 6.2.3: Technologies Used in Proposed Work



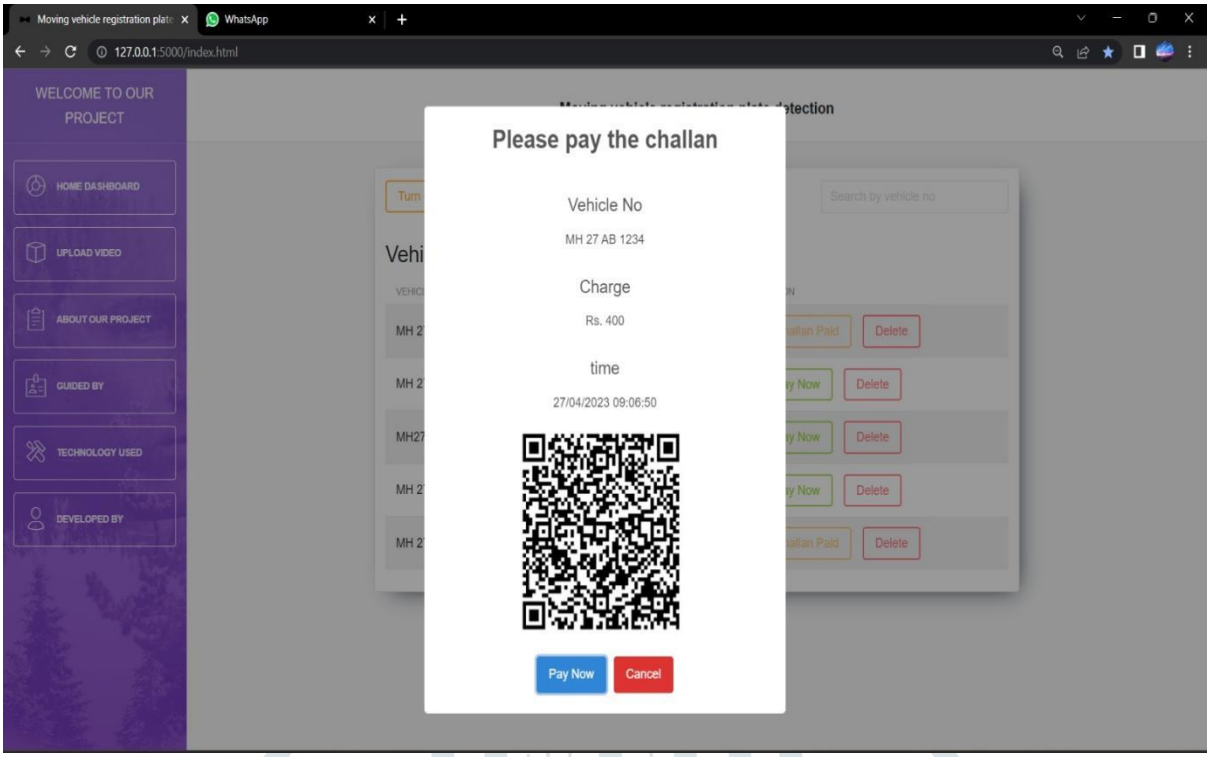
Screenshot 6.2.4: Project Details



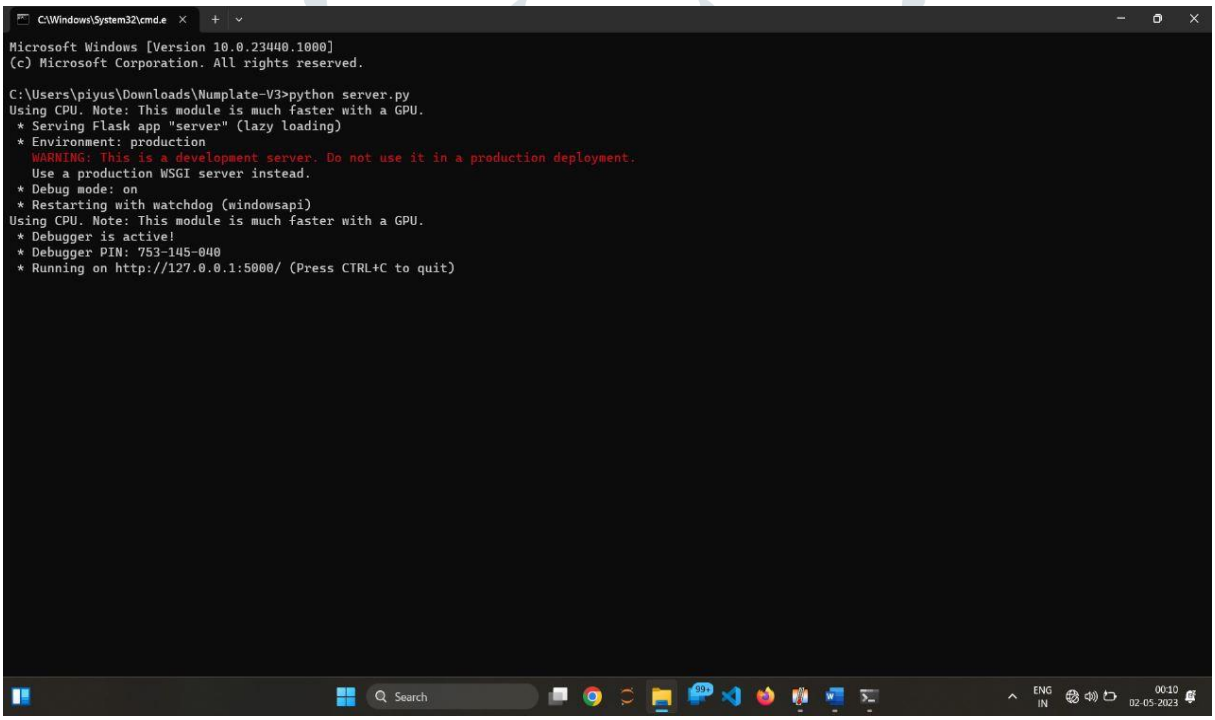
Screenshot 6.2.5: Project Developer



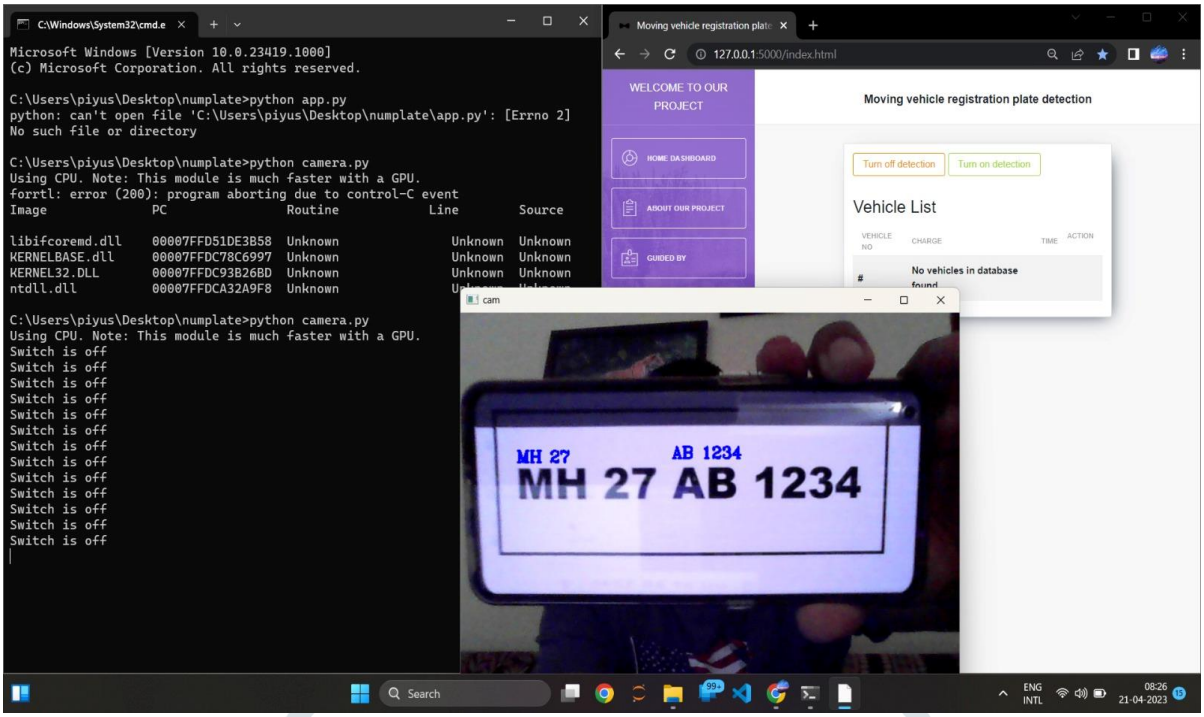
Screenshot 6.2.6: Complaint History



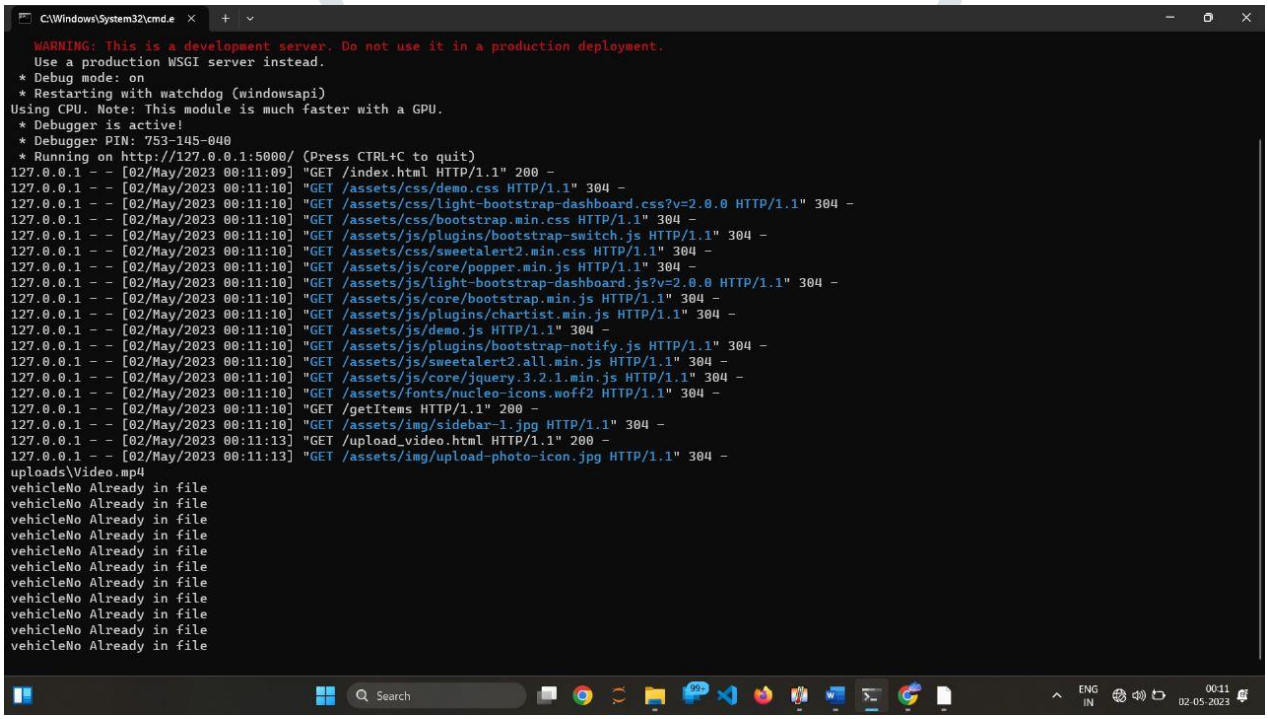
Screenshot 6.2.7: QR Facility for paying Challan



Screenshot 6.2.8: Sever Started



Screenshot 6.2.9: Number Detected



Screenshot 6.2.10: Vehicle Number Detected

VII. Advantages And Disadvantages

Advantages

1. **Enhanced Efficiency:** The automated nature of the system eliminates the need for manual identification and processing of registration plates, leading to significant time savings and improved operational efficiency. It allows for real-time detection and prompt issuance of e-challans, ensuring swift enforcement actions.
2. **Improved Accuracy:** By utilizing computer vision and machine learning algorithms, the system achieves a high level of accuracy in detecting and recognizing registration plates. This reduces the chances of errors or misinterpretations that may occur in manual identification methods, ensuring more reliable enforcement of traffic regulations.
3. **Timely Enforcement:** The real-time capabilities of the system enable immediate identification of violators, allowing for prompt enforcement actions. This helps in deterring traffic offenses, improving overall compliance, and enhancing road safety.
4. **Reduced Administrative Burden:** The electronic issuance of e-challans eliminates the need for manual paperwork, reducing administrative burdens for law enforcement agencies. This saves time, resources, and streamlines the process of managing and maintaining records of traffic violations.
5. **Integration with Existing Databases:** The system can be integrated with existing databases of registered vehicles, enabling seamless access to relevant information such as vehicle ownership details, valid registrations, and previous violations. This integration facilitates more comprehensive monitoring and analysis of traffic offenses.

Disadvantages

1. **Initial Implementation Costs:** The deployment of a moving vehicle registration plate detection and e-challan system requires a significant upfront investment in terms of hardware, software, and infrastructure setup. This can pose a financial challenge for some authorities or jurisdictions, particularly those with limited budgets.
2. **Technological Limitations:** The effectiveness of the system is contingent upon the reliability of the underlying technologies, such as computer vision algorithms and OCR capabilities. Factors such as adverse weather conditions, poor lighting, or obscured registration plates may affect the system's performance and accuracy.
3. **Privacy Concerns:** The use of surveillance cameras and automated detection systems raises privacy concerns. There is a need to ensure that the collected data is handled securely and used solely for the purpose of traffic enforcement, without compromising individuals' privacy rights.
4. **Vulnerability to Hacking or Manipulation:** Like any digital system, there is a risk of potential cyber-attacks or manipulation of the moving vehicle registration plate detection and e-challan system. Adequate cybersecurity measures must be in place to safeguard the system against unauthorized access or tampering.
5. **Dependence on Infrastructure**: The effectiveness of the system is reliant on a robust infrastructure, including a network of surveillance cameras, data connectivity, and power supply. In regions with inadequate infrastructure, the system's performance may be compromised.

While the advantages of a moving vehicle registration plate detection and e-challan system outweigh the disadvantages, it is crucial to address the potential drawbacks through proper planning, implementation, and ongoing maintenance to ensure its successful deployment and operation[6].

VIII. Applications

Moving vehicle registration plate detection has numerous applications in various fields. Here are some of the key applications of this technology:

1. **Traffic management:** One of the most common applications of moving vehicle registration plate detection is in traffic management. By automatically capturing and analyzing the number plates of vehicles, traffic authorities can monitor traffic flow, identify and track speeding vehicles, enforce traffic laws, and respond to accidents and emergencies quickly.

2. **Toll collection systems:** Moving vehicle registration plate detection can be used in toll collection systems to automate the process of collecting tolls. The technology can capture the number plates of vehicles as they pass through toll gates, and automatically deduct the toll charges from the vehicle owner's account.

3. **Parking management:** Moving vehicle registration plate detection can be used to automate parking management systems. By capturing the number plates of vehicles as they enter and exit parking areas, the technology can enable real-time monitoring of parking spaces and reduce traffic congestion.

4. **Law enforcement:** Law enforcement agencies can use moving vehicle registration plate detection to track and identify vehicles involved in criminal activities. The technology can be used to automatically scan number plates and compare them with a database of registered vehicles to identify vehicles of interest.

5. **Border security:** Moving vehicle registration plate detection can be used to improve border security by monitoring the movement of vehicles across borders. By automatically capturing and analysing number plates, authorities can identify suspicious vehicles and monitor their movement.

6. **Environmental monitoring:** Moving vehicle registration plate detection can be used to monitor and track vehicles that emit excessive amounts of pollutants, enabling better enforcement of environmental regulations.

7. **Smart city infrastructure:** Moving vehicle registration plate detection can be integrated with smart city infrastructure to monitor and track vehicles, enabling better traffic management, improved security, and real-time monitoring of air pollution levels.

8. **Autonomous vehicles:** Moving vehicle registration plate detection can be used by autonomous vehicles to automatically identify and track other vehicles on the road, enabling safer and more efficient self-driving systems[7].

IX. CONCLUSION

In conclusion, the moving vehicle registration plate detection and e-challan system presents a promising solution for enhancing traffic management and enforcement processes. By leveraging computer vision, image processing, and machine learning technologies, this system automates the identification of registration plates in real-time, leading to efficient and accurate issuance of e-challans for traffic violations.The advantages of this system are numerous. It improves operational efficiency, reduces manual errors, and enables prompt enforcement actions, contributing to enhanced traffic flow and road safety. The electronic issuance of e-challans reduces administrative burdens, eliminates paperwork, and promotes a more sustainable and eco-friendly approach to traffic management. Furthermore, the integration with existing databases allows for comprehensive monitoring and analysis of traffic offenses.However, there are also potential challenges to consider. Initial implementation costs, technological limitations, privacy concerns, and the risk of cyber-attacks or system manipulation require careful attention during the planning and deployment phases. Additionally, the availability of robust infrastructure is crucial for the system's optimal performance.
Despite these challenges, the moving vehicle registration plate detection and e-challan system holds great potential in revolutionizing traffic management and enforcement. By automating the process of registration plate detection and issuing e-challans, it improves efficiency, accuracy, and compliance with traffic regulations. Ultimately, the successful implementation of this system can contribute to safer roads, reduced traffic congestion, and a more organized transportation environment.

X. FUTURE SCOPE

Moving vehicle registration plate detection technology has come a long way, and its future scope is vast and promising. Here are some of the key areas where this technology is expected to evolve in the future:

1. **Improved accuracy and speed:** Moving vehicle registration plate detection is expected to become more accurate and faster in the future, enabling real-time processing of large volumes of data.

2. **Integration with other technologies:** Moving vehicle registration plate detection is expected to be integrated with other technologies, such as machine learning, artificial intelligence, and big data analytics, to enable more advanced applications.

3. **Advanced surveillance:** Moving vehicle registration plate detection is expected to be integrated with surveillance cameras, enabling the automatic tracking and monitoring of vehicles in real-time.

4. **Autonomous vehicles:** Moving vehicle registration plate detection technology is expected to play a significant role in the development of autonomous vehicles. It can be used to identify and track other vehicles on the road, enabling better navigation and safety.

5. **Smart cities:** Moving vehicle registration plate detection is expected to play a vital role in the development of smart cities. It can be used to monitor and regulate traffic, reduce congestion, and improve air quality.

6. **Improved security:** Moving vehicle registration plate detection is expected to be used in the development of advanced security systems, enabling the automatic detection and tracking of suspicious vehicles.

7. **Environmental monitoring:** Moving vehicle registration plate detection is expected to be used in the development of environmental monitoring systems, enabling the automatic detection and tracking of vehicles that emit excessive amounts of pollutants.

8. **Improved toll collection systems:** Moving vehicle registration plate detection is expected to be used in the development of improved toll collection systems, enabling the automatic deduction of toll charges from vehicle owner's accounts.

REFERENCES

[1] K. R. K. Reddy, S. Ramana, and P. S. Kumar, "Automated Traffic Management System using License Plate Recognition and E-challan Generation," International Journal of Computer Applications, vol. 146, no. 9, pp. 1-4, 2016.
[2] R. M. Swamy, V. R. Reddy, and N. B. V. R. Bhavanam, "License Plate Recognition using Neural Networks and E-Challan Generation for Traffic Rule Violations," International Journal of Innovative Research in Computer and Communication Engineering, vol. 3, no. 7, pp. 6675-6682, 2015.

[3] T. K. Panda, K. C. K. Rao, and P. S. S. S. M. M. Rao, "License Plate Recognition for Traffic Management and E-challan Generation," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 4, no. 7, pp. 662-666, 2014.
[4] S. M. Alomari, M. F. Tappen, and M. Alkhatib, "A survey on license plate recognition systems," Journal of Applied Research and Technology, vol. 16, no. 5, pp. 413-428, 2018.
[5] A. M. Lopez-Mendez, A. J. Rivera-Alba, and R. Garcia-Cabrera, "License plate detection and recognition: A review," IEEE Latin America Transactions, vol. 14, no. 4, pp. 1752-1759, 2016.
[6] X. Li, X. Shen, and Z. Li, "Vehicle license plate recognition using deep learning techniques: A review," IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 3, pp. 869-886, 2018.
[7] Z. Liu, Y. Sun, and W. Zheng, "License plate detection and recognition using deep learning: A review," IEEE Access, vol. 7, pp. 133211-133231, 2019.
[8] Jameson, H. S. Abdullah, S. Norul, A. N. Ghazali, N. Nur, and N. A.Zamani, "Multiple Frames Combination Versus Single Frame Super Resolution Methods for CCTV Forensic Interpretation," Journal of Information Assurance & Security, vol. 8, 2013.
[9] B. Zitova and 1. Fiusser, "Image registration methods: a survey," Image and Vision Computing, vol. 21, no. II, pp. 977- 1000, 2003.
[10] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," Signal Processing Magazine, IEEE, vol. 20, pp. 21-36, 2003.