AUTOMATIC INDIAN CAR NUMBER PLATE RECOGNITION

Kadiri Anish Kumar, Department Of Electronics and Communication Engineering Andhra University College Of Engineering, Visakhapatnam : anishkumarkadiri3@gmail.com
Dr.S.Aruna Assistant Professor, Department Of Electronics and Communication Engineering Andhra University College Of Engineering, Visakhapatnam : aruna9490564519@gmail.com
Dr. K. Srinivasa Naik Associate Professor, Department Of Electronics and Communication Engineering Vignan's Institute of Information Technology, Visakhapatnam
Email: nivas97033205@gmail.com

Abstract—The Indian Auto Industry is growing at a very high rate and is soon going to be the World's third-largest automobile market in the world. This growth led to the rise in the vehicle registrations in the country. Each registration number is unique and also can be used to identify a vehicle.

The fake registration numbers and also multiple vehicles having same registration number also are a big threat to the traffic monitoring system. Some Authorized Institutions and Organizations only allow authorized vehicles. A robust system is needed for identifying the number plates on vehicles.
Number plate detection is a method used to extract the letters, digits from the number plates of vehicles. Various edge detection techniques, Image processing techniques, Image restoration and Enhancement Techniques are used.

Keywords— Python, Opencv, Contours, Image Processing, Pytesseract.

I. INTRODUCTION

In present days, the transportation system has undergone several changes as the vehicles being used are increased for both commercial and private use. The respective road transport authority of each state makes sure regarding the vehicles registration and authencity in vehicles. Number plate is a major part in identifying the vehicle particulars such as registered state, past history, owner identification. In India there are standards while registering the number plate. The length is capped at 10 digits and varies between 8 to 10-digit plates and the formats are discussed in further sections.

These are some people that reluctantly drive and there are many people who use fake registration numbers. There are numerous cases being reported which are related to traffic offense. A huge rise in the vehicle theft cases has also been seen in current world. There should be a live detection mechanism for detecting number plates which can be used to easily identify the vehicle. There are many researches which are done in detecting the number plate. This work provides a detection mechanism that can use video based data in order to detect the vehicle number plate. Based on the current traffic situation, the proposed work can be used to identify the vehicles. The vision system helps to analyze current traffic violations, identifying vehicles that are suspicious. The Combination of python and computer vison has potential to identify frames in video by image processing and image enhancement techniques. By analyzing the structure of vehicles, and also the position at which the system is placed the detection is done. The frame work uses Python, OpenCV, and Pytesseract.

II. RELATED WORK

To match with ongoing research and observe the current moments in the detection of vehicles and recognizing the number plate. The work has been done on Python, OpenCV. The video taken can be from video camera or prerecorded video. The video based data can be read as frame by frame image and the challenge lies in detection from frames of video which change as 24 frames per second and the transition from one frame to other makes it difficult for detecting the number plate. There is no object detection done and the number pate is directly detected based on the contours detected from the image frame and this process is done on Jupyter Notebook.
The image frame and the detected plate are taken from the video and stored in the local storage for future reference. This system can be further developed in order to improve and implement using high-infra-red cameras. The camera used while experimenting belongs to the smartphone camera.

III. SOFTWARE

A. OPENCV2

OpenCV developed by Intel came out in early 2000.

OpenCV is an open-source cross-platform library used to develop machine learning, image processing, and real-time computer vision applications. Its primary focus lies on the concepts related to image processing, video capture, and analysis including features like object detection and object recognition. OpenCV supports various programming languages like Python, C++, Java, etc. It can process ,analyze pictures and videos to identify various subjects, faces, and text. OpenCV is useful when integrated with various libraries, such as Numpy, Matplotlib, etc.

B. PYTESSERACT

Tesseract was developed by Hewlett Packard Labs as proprietary software. In 2005, in collaboration with the University of Nevada, Las Vegas HP, started open-sourcing Tesseract. Many open source contributors notably google made major contributions to developing Tesseract.

Tesseract with version 3.x started supporting many image formats and gradually added a large number of scripts (languages). Initially, Tesseract worked using traditional computer vision algorithms. Later Deep Learning based methods surpassed traditional machine learning techniques and the gap further increased due to the advancements in deep learning in many areas of Computer Vision.

C. PYTHON

Python is a popular programming language used for web and software development, and system scripting. and mathematics. Python is an object-oriented, cross-platform working language that can be used on different platforms such as Windows, Mac, Linux, Raspberry Pi, etc. Python has a syntax similar to the English language that can be understood by researchers when compared to other Complicated syntax languages. Python's simple syntax allows developers to write programs with fewer lines than some other programming languages. Python has been a favorite tool among researchers when it comes to using OpenCV.

IV. METHODOLOGY

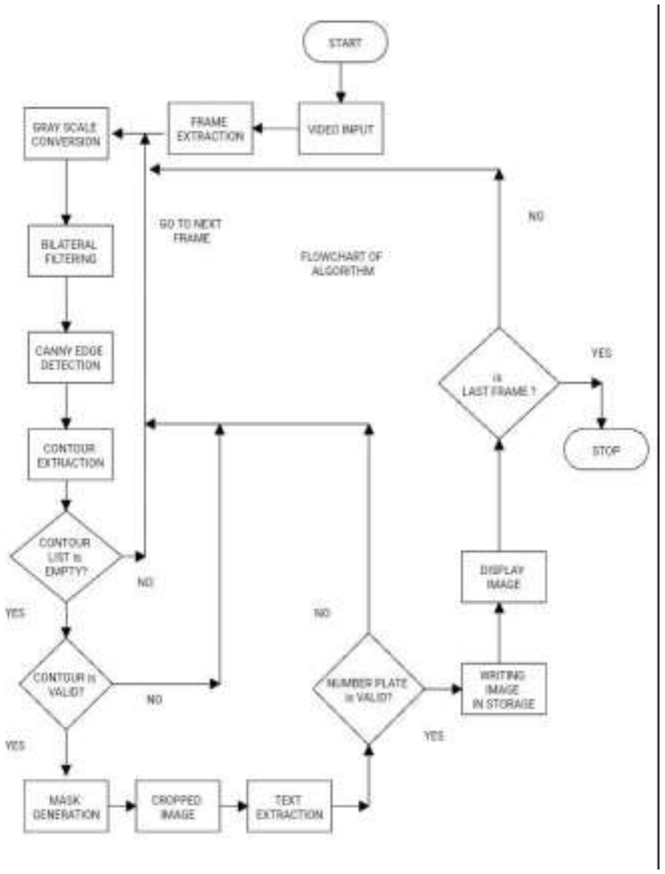
The suggested algorithm begins with entering input which is a video into the framework. The frame-by-frame images are extracted from the video. Then each frame is converted to a grayscale image. Then the grayscale images are given to the next stage where bilateral filtering is applied and the filtered image is generated. Then the canny edge detection is performed for detecting the contours on the frame extracted from the video.

If contours detected are empty we cannot proceed to further stages of the framework so all the next steps are skilled and the next frame is processed. These stages repeat until all the frames of the video are processed.

From the list of contours, each contour is approximated to be a four-sided figure, and if the condition is satisfied then the location of the contour is passed on to the next step. If the approximation condition fails then the next steps are skipped and the next contour item in the list is processed to the next stage. If a contour is found suitable then in the next stage a mask (preferably black (0)) is generated and the location of the contour region is inverted on the mask (to white (1)) which creates a new mask. Then the new mask generated is used to Perform the bitwise operation on the frame and the number plate is extracted from the frame.Then the extracted image is cropped only on the regions where the number plate is visible. This cropped image of the number plate is processed to the next stage which contains pytesseract and the image text is taken out. The number plate (image text) is stored in the new database. The text is processed and if the text is as per standards of Indian number plate formats.

If the condition is satisfied then the number plate image is saved in the local database and also the car number plate is displayed on the screen.The number plate text is passed to the next stage in which the extracted text is compared to the database which contains the state information. Then the state in which the car is registered is displayed on the screen.

A. Proposed model



B. Number Plate model in India

a) Ten-digit Number Plate:

Here is an example of 9 digit number plate which consists of **XXNNXX NNNN** and here the **DL** (Delhi - State Code), **8C** (North West Delhi – District Code), **AC** (Series Code), **4194**(Vehicle Registration Number).

Fig 1. Ten digit number plate

b) Nine digit Number Plate:

Here is an example of 9 digit number plate that consists of **XXNN X NNNN** and here the KA (Karnataka - State Code), **19**(Mangalore – District Code), **P** (Series Code), **8488**(Vehicle Registration Number).

Fig 2. Nine digit number plate

c) Eight-digit Number Plate:

Here is an example of 8 digit number plate which consists of **XXNN NNNN** and here the **CH** (Chandigarh – State Code), **04**(Chandigarh – District Code), **1454**(Vehicle Registration Number).

Fig 3. Eight digit number plate

C. Working of Algorithm

The video taken was used and the video sample frame is shown below and this frame is processed further.

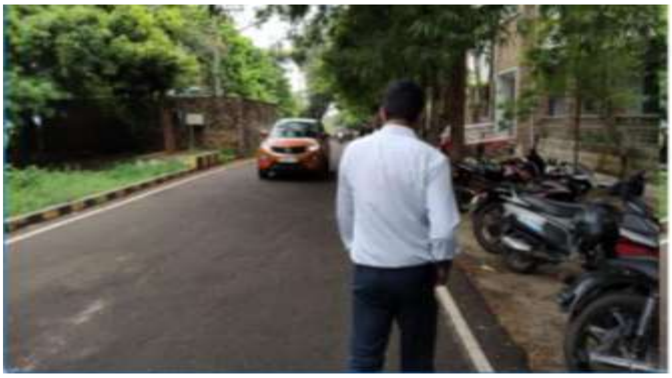


Fig 4. A sample frame from the video

This Image frame is extracted from the video and the frame is further processed to the next stage and the grayscale image is produced.

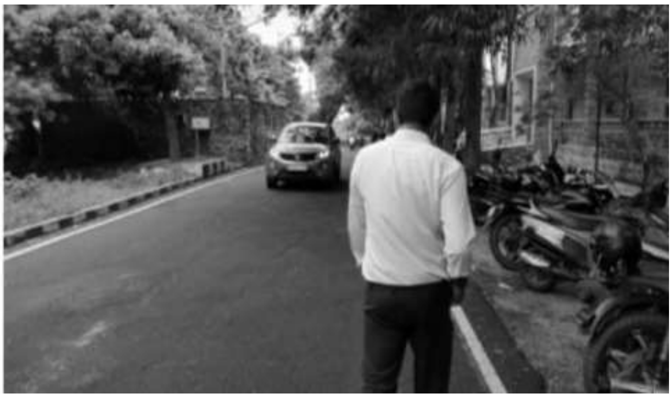


Fig 5. Gray scale converted Image frame

This grayscale image generated is sent for further stages for processing, a bilateral filter is used in order to filter the noise and preserve the edges for better edge detection. Then later the image frame is sent into the canny edge detection for detecting the edges. After all the edges are detected the contours are extracted from the image and the extracted contours are limited only top 30-50 depending on the requirement.

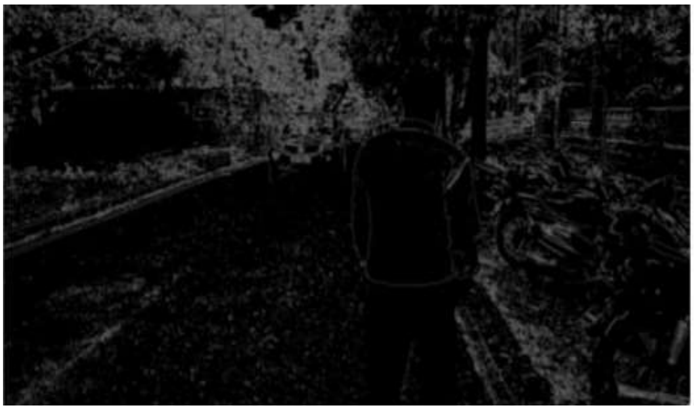


Fig 6. Image frame after canny edge detection

After the contours are found the suitable contour is chosen and the number plate is extracted and it is sent to OCR for detection and the number plate text is extracted. The text is sent for detecting a match in the database for matching state information and it is displayed on the screen. The number extracted should be in the prescribed number plate format I.e. 10 digit, 9 digit, and 8 digit number plate.

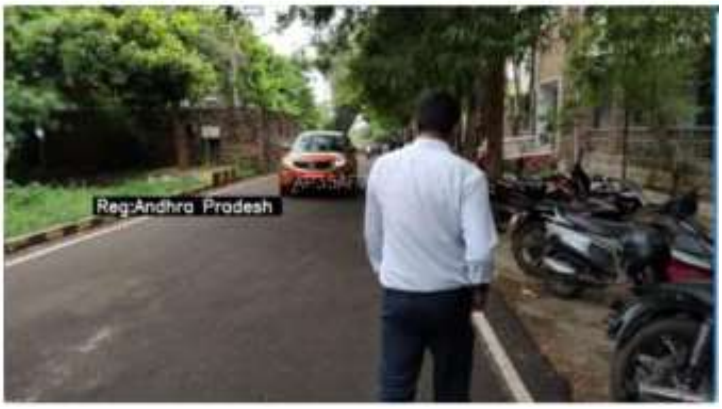


Fig 7. The registration state being shown on the screen

The state information and the vehicle registration number are displayed on the screen and it continues for the next frames also the Number plate detected is extracted out as output and is stored in a preferred database.

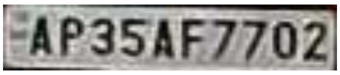


Fig 8. The detected number plate

After the plate is detected and stored the image can be used for further reference, and the accuracy can be given as 100%. Depending on the increase in the vehicles detected the parameter accuracy comes into the picture which is also found for comparison purposes. Consider another sample that contains two different vehicles and the detection of both can be seen. Another sample Image frame can be seen with two different cars present that can be used as a reference for the identification of number plates. The input frames are shown below and the sample frame one is seen below.

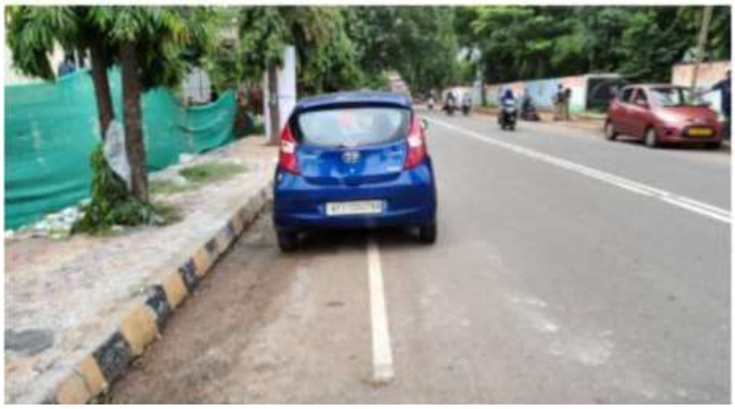


Fig 9. A sample frame from the video

The two cars can be seen in the frame and the frame is a part of video input to the frame work. The gray scale image is seen below.

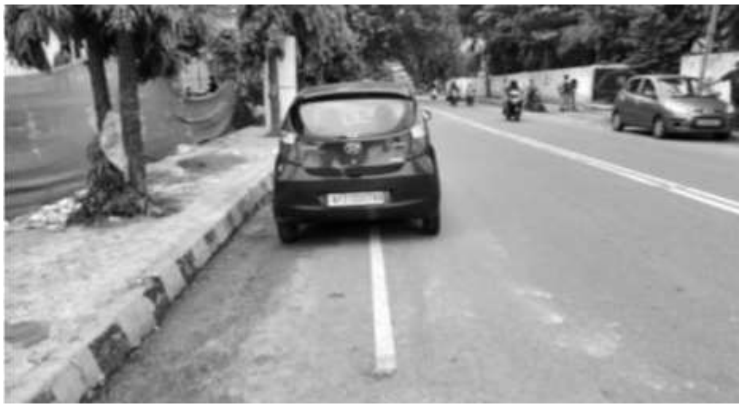


Fig 10. Gray scale converted Image frame

The generated grayscale image is sent for bilateral filtering and also the canny edge detection which can be seen in the below image where all the edges are seen.

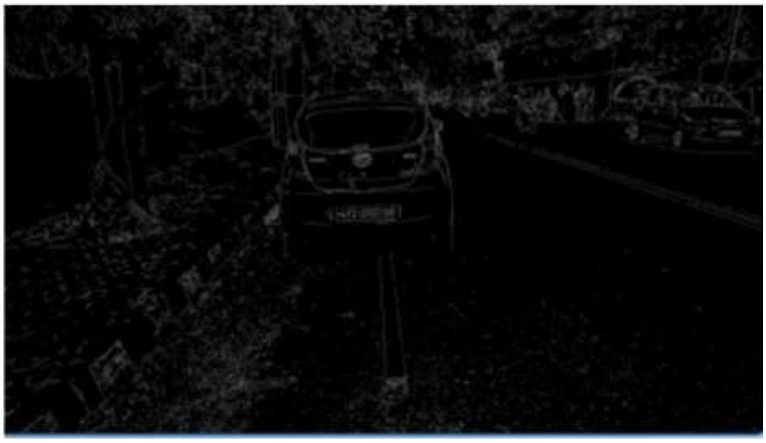


Fig 11. Image frame after canny edge detection

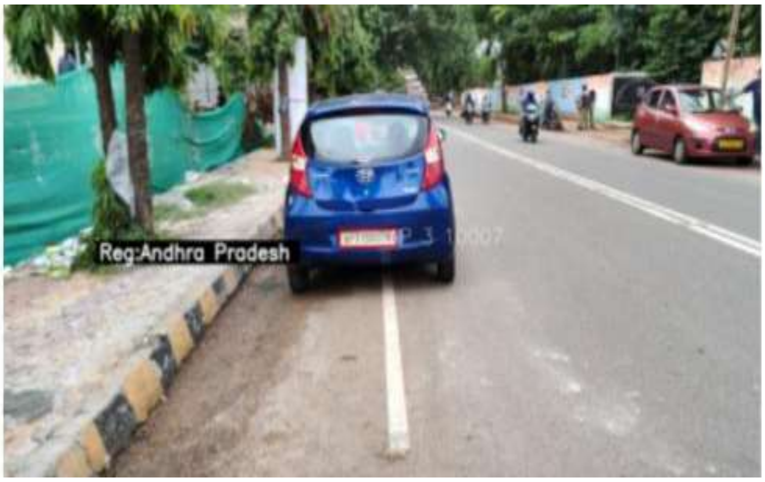


Fig 12. The registration state being shown on the screen

The plate detection can be seen in the image frame and the registration number and state are also shown.

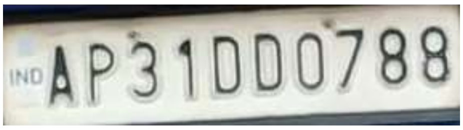


Fig 13. The detected number plate

The plate detected count is one and the accuracy is 100%. Now considering the second sample image frame from the video we can see the below input image frame.

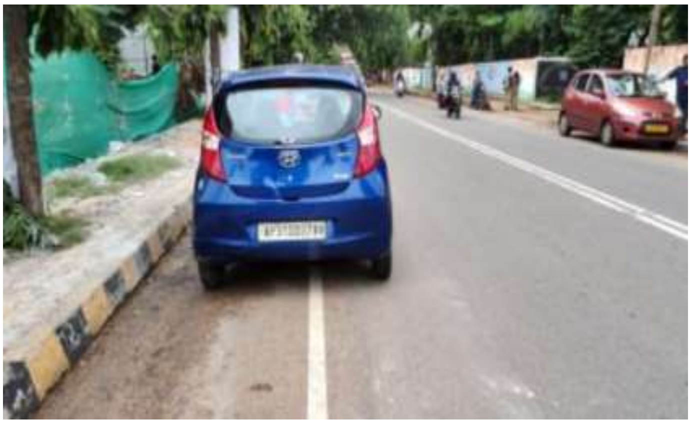


Fig 14. A sample frame from the video

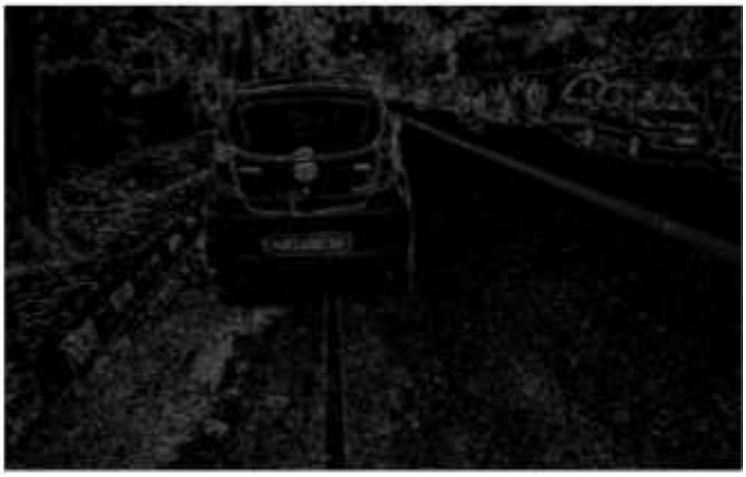


Fig 15. Gray scale converted Image frame

The Blue car can be taken as a reference and also a maroon car is also seen in the frame, this image is also processed and the gray-scaled version is seen below.

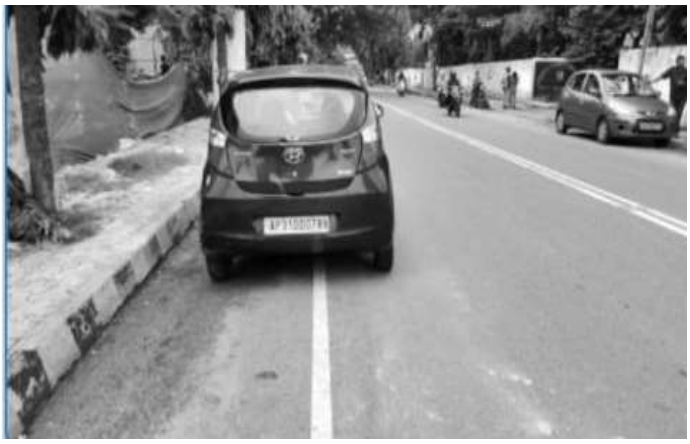


Fig 16. Image frame after canny edge detection

In further stages, it is sent for bilateral filtering and also the canny edge detection which can be seen in the below image where all the edges are seen.

The canny edge detected image acts as an input and the number plate is detected in the next stage.

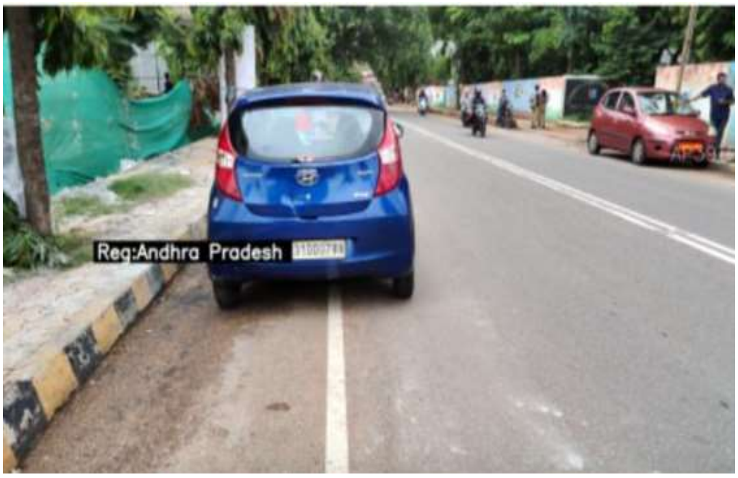


Fig 17. The registration state being shown on the screen

The second car number plate is highlighted and can be seen in this image frame. The extracted image of the number plate is seen below.

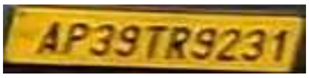
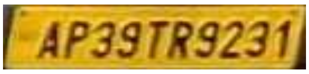
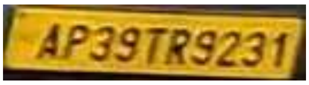
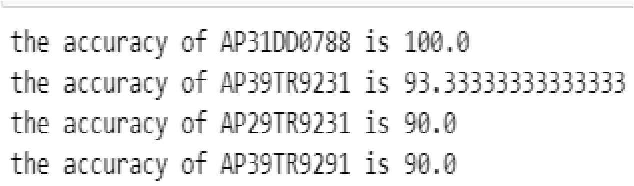
Detected/Extracted	Detected Plate Number
	AP39TR9231
	AP29TR9231
	AP39TR9291

Fig 18. The extracted number plates and the number plate text.

This approximation error has occurred due to various factors which cannot be avoided like the plate being far from the camera and also the frame-by-frame transition of the video also might have resulted in a problem. The camera used is generic and not a special purpose system for capturing the Images. So an accuracy system is proposed in this framework to compare the number plate texts that are extracted and the percentage of matching is can be taken as a parameter of efficiency. The comparison among the text is done on basis of the number plate structure being used in India. As the only common parts of two number plates can only the first two characters depict the state of registration of the car? A second part is a number that has a respective meaning for every state and is not at all common but the vehicles can be from the same district or area so the common characters count rises to 4 out of 10. The unique code which is present after these two identifies is specifically or uniquely designed by the state RTOs and the information is confidential as to how the numbering is designed.

So the limit is set to 5 out of 10(max) characters. The number plate text with less than 5 matching characters is set to 100% accurate and is stored and the plate which is above 5 matching characters is processed and the plate text finding accuracy is found out.



```
the accuracy of AP31DD0788 is 100.0
the accuracy of AP39TR9231 is 93.33333333333333
the accuracy of AP29TR9231 is 90.0
the accuracy of AP39TR9291 is 90.0
```

Fig 19. The accuracy calculation in case of multiple plates being shown on the screen

This accuracy is found only based on the data or number plate text extracted from that video only.

V. RESULT

The images labeled as Fig 8, Fig 13, and Fig 18 show the extracted output generated by processing the input video data and the image frame that is extracted from the video. Figure 8 is the image extracted from the video that contains only a single car that can be seen and also the accuracy for that car is 100% based on the data extracted from the frames of the video. The fig 13 is also the output of an image frame extracted from the video that contains two cars, the plate extracted is one and the text generated is also one so the accuracy as shown in fig 18 is 100%, and also for the second car the detected images are similar but the text generated is not same due to the errors that arise due to environmental conditions and also depend on the equipment capabilities. The respective accuracies are shown in fig 18 which depends only on the data obtained from the video input. The test failed only on videos that do not have a sufficient amount of light and also in places where they were hardware limitations.

VI. CONCLUSION

In this paper, an efficient method for the detection of the number plate in both video format and the live camera is implemented. The frames were tested with a bilateral filter and Canny edge detection method that is proven to be efficient in terms of the framework presented in this paper. The frames are captured when the text is recognized in the video frame and are stored in local storage in a folder that is titled with the time stamp of the test run. The cropped images presented in this paper are also obtained from the folder data. The image converted to text is stored in the form of a String. The pytesseract OCR was used to detect the text from the cropped images. Our code worked properly for all the data taken from the smartphone camera and any improvement in the hardware can improve the results obtained in this paper. The test failed only on videos that do not have a sufficient amount of light and also in places where they were hardware limitations.

VII. REFERENCE

[1]. A. S. Mohammed Shariff, R. Bhatia, R. Kuma, and S. Jha, "Vehicle Number Plate Detection Using Python and Open CV, "International Conference on Advance Computing and Innovative Technologies in Engineering in 2021

[2]Wengang Zhou, Houqiang Li, Yijuan Lu, Qi Tian, "Principal Visual Word Discovery for automatic License Plate Detection," Image Processing in IEEE Transaction.

[3].Yogiraj Kulkarni, Shubhangi Bodkhe, A. Kamthe, and A.Patil, " Automatic number plate recognition for motorcyclists riding without a helmet " International Conference on Current Trends towards Converging Technologies in 2018.

[4]."Automatic car number plate recognition," International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS) in 2017.

[5].A. Kashyap, B. Suresh, A. Patil, S. Sharma, and A. Jaiswal, " Automatic Number Plate Recognition " International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)in 2018

[6].Pratiksha Jain, Neha Chopra, and Vaishali Gupta, " Automatic License Plate Recognition using OpenCV", International Journal of Computer Applications Technology and Research (IJCATR).

[7]Reza Azad and Hamid Reza Shayegh, " New method for optimization of license plate recognition system with the use of edge detection and connected component," ICCKE in 2013