

# Vehicle Number Management using Hbase



Chetan Pandey, Amit Juyal, Poonam Verma, Kamred Udham Singh

**Abstract:** *Intelligent Monitoring and Recording System (IMRS) is being used in many fields like Aviation Traffic control, Transportation, Real Estate, Medical Science and more. One of the IMRS system is Vehicle Traffic on roads. Intelligent Transportation System (ITS) is an important part of IMRS which is used to collect and analyze the statistics related to the Vehicle Traffic. This paper gives an insight of the ITS by using open source tools which are easy to implement. Large datasets of vehicle information has been taken into the experiments of this paper. Utilizing different digital image processing techniques, we have extracted vehicle number from the number plates. Technically, this paper is based on ITS in which two main features has been designed, first is to store Vehicle related information in the Hbase. Another feature is to retrieve the data, on the basis of the vehicle number, from the database giving the details of the vehicle including Road Tax, Insurance and Stolen status and also inform the possessor about the invalidity of Insurance, Registration Certificate (RC) and License. Time and accuracy are two challenges in performing the IMRS in a real life scenario. It has been observed that the proposed system is 27% faster than Automatic Number Plate Recognition (ANPR) on the basis of two comparative parameters i.e. Precision and Recall*

**Keywords :** Hadoop, HDFS, HBase, OpenCV, OCR, Tess4J

## I. INTRODUCTION

Detection of Vehicle Number plays a very important role in Intelligent Monitoring and Recording System (IMRS) [6][16]. Intelligent Monitoring covers supervision of the target system like road traffic, aviation traffic, the stock market, some parking areas and more. This is being achieved by using cameras, sensors e.g. proximity sensor or any suitable scientific tools. Recording System implies a collection of images, videos or the outcomes of tools used for monitoring. This paper focuses on the Vehicle Traffic [16] and will majorly focus on the extraction of Vehicle Number from the Vehicle and storing it in HBase [13], an NoSQL database, alongwith some important information of the vehicle like name of the person to which vehicle is registered, contact, address and more.

Manuscript published on November 30, 2019.

\* Correspondence Author

**Chetan Pandey\***, Graphic Era Hill University, Dehradun, Uttarakhand, [schetanpandey@gmail.com](mailto:schetanpandey@gmail.com)

**Amit Juyal**, Graphic Era Hill University, Dehradun, Uttarakhand, [amitjuyal26@gmail.com](mailto:amitjuyal26@gmail.com)

**Poonam Verma**, Graphic Era Hill University, Dehradun, Uttarakhand, [poonamddn18@gmail.com](mailto:poonamddn18@gmail.com)

**Kamred Udham Singh**, Graphic Era Hill University, Dehradun, Uttarakhand, [kamredudhamsingh@gmail.com](mailto:kamredudhamsingh@gmail.com)

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

In India, basically there are two types of license plates, one is black characters in a white plate (private vehicles like Bike and Car) and another is the black characters in a yellow plate (for commercial uses like Taxi), which makes it challenging for capturing images or making videos of these vehicles. White does not make much trouble but yellow color is difficult to handle since when some light fall over it, it becomes too much bright making the number to appear dull in the image. Also, style of writing numbers over license plate is different i.e. some plates have the number in one line and some have two lines in the license plate. However, the recognition task becomes challenging if the color of vehicle number is very similar to the background since some uses multicolor in vehicle numbers. The HBase [13] used here is a column-oriented key-value data store which work with Hadoop [15] Distributed File System (HDFS) [14]. HBase has fast records lookups and updates, for this reason it is widely used for the real time applications. Considering this feature as an advantage, in this paper we focused on how to use HBase for storage as well as for some real time applications. Our first task is to capture Vehicle number plate by using good quality of digital cameras like DSLR based. However, due to human error and some natural factors, it is highly possible that many images are not clear, which finally makes it difficult to extract vehicle number from them. Also, the vehicle owner may use a different style of writing vehicle number and also different country have different syntax of vehicle number plate - all these factors create disturbance and noise in the collection process of the vehicle number. This paper will discuss a user-oriented approach to remove noise from the image after applying digital image processing techniques [4]. The basic techniques are to convert RGB image to Gray Scale Image, Edge Detection, Dilation and Erosion. If required De-blurring algorithms is used for correction of blurring and defocusing of vehicle images. After that Character Segmentation [7] is used and finally license number is extracted from a license plate image using Optical Character Recognition (OCR) [3] [5]. An HBase [13] dataset of Vehicle numbers has been created containing basic details of the vehicle along with Road Tax, Insurance and stolen status. Once the number is extracted, the Java based Application designed for the purpose checks for the number in the dataset if found then all the details are displayed along with a warning bell in case of theft status yes, insurance due and road tax due a warning message is displayed.

## II. RELATED WORK

Detection of Vehicle Number from Vehicle Image is although too old research area but still due to its importance and value in other area of research, like Stolen car detection, Automated parking system and more, it still an interesting and evergreen area of research.

## Vehicle Number Management using Hbase

There are many ways to extract license's character into some editable text format like some paper shows the use of Neural Network [1], Support Vector Machine [1], OCR [12], Matlab [6], Hough Transform, Soft Computing and more. Like in paper Learning-Based Approach For License Plate Recognition [1] published in the year 2000 by K.K.Kim, K.I.Kim, J.B.Kim and H. J.Kim, uses neural network for image segmentation and support vector machine for character recognition. Another paper published in the year 2004 named Automatic License Plate Recognition [11] by Shyang-Lih Chang, Li-Shien Chen, Yun-Chung Chung and Sei-Wan Chen make use of Soft Computing Techniques for the detection of vehicle number.

Focusing on the pre-processing [10], in the year 2014, a paper [3] is published in which author suggested how to extract numbers from the number plate using edge detection and dilation especially. In 2014 another paper [4] preferred free and open source tools like OpenCV over Licensed based tools like MatLab by using Python programming to form Automatic License Plate Recognition system. A research paper [12] published in the year 2014 suggested a design of extraction of metadata of attributes like license plate number, vehicle type and more and incorporate the idea of creating a separate metadata using graph concept and make use of it for metadata correction purpose. Many research papers show some automatic technology of detecting numbers from vehicle images. One of the paper [4] is published in the year 2014 author designed an efficient automatic vehicle identification system. There are some research papers focusing on some different technology like Optical Character Recognition (OCR) to not only detecting license number but also storing them in editable text format at the same time. One of such paper [5] published in the year 2015 which presented a methodology of extracting the number from the captured vehicle image by using Java OCR Libraries. In 2016 there is another research paper [11] which focuses on the application of checking whether a vehicle is registered or not by using some pre-stored datasets of vehicle related information. Some authors work more over it by storing vehicle related data in NoSQL database for some real time application like in 2017 a

paper [12] discussed about parallel massive data loading over HBase. According to this paper, the ETL (Extract-Transform-Load) process and then finally the processing of these data is effectively implemented using HBase.

In section III authors explain about the dataset and the methodology of vehicle number extraction along with its algorithm. IVth section covers how the proposed framework will work in ITS. This section also discussed an algorithm of managing and processing of vehicle related information in HBase database. In section V authors examine the emphasis of the proposed framework in terms of Time, Precision and Recall.

### III. PROPOSED FRAMEWORK

Let's first discuss about HBase [13] which is used here for back-end processes and is widely used by developers for developing surveillance related applications. HBase is based on Hadoop's HDFS [14] so first we need to install Hadoop [15] on the system and then finally HBase is installed within the Hadoop environment. HDFS allows the parallel processing by breaking the information into separate pieces distribute them to different nodes either in a cluster or multi cluster. Another important feature which makes it highly accepted file system is that it makes copies of data elsewhere within a cluster, which allows processing of data even if one copy of data crashes.

Here in this paper the methodology suggested is based on single cluster and uses a big data set of vehicle including the information like Vehicle Number, Registration Date of Vehicle, Name of the person used for Registration of vehicle, Address, Contact Number, Date of last road tax deposited, Date of last insurance and finally Stolen status in Boolean form i.e. 0 if not stolen and 1 if vehicle is stolen. Fig. 1 showing the screenshot of the dataset used for this suggested methodology.

	A	B	C	D	E	F	G	H
1	Veh_Num	Regist_Date	Name	Address	Contact	Last_RTax_Dep	Last_Insurance	Stolen_Status
2	UK07G1234	12-07-2001	S.K.Sharma	bhallawala road racecourse Dehradun	895658452	14-02-2017	12-12-2016	0
3	UK07G1258	12-08-2002	M.P.Yadav	ladpur Dehradun	854265985	14-03-2017	12-10-2016	1
4	UK07G5214	30-08-2009	H.K.Verma	new colony abc street haridwar	986584575	24-01-2016	27-01-2017	0
5	UK07H5468	22-08-2008	C.B.Papnai	uccha pull haldwani	785986231	27-06-2017	17-03-2017	0
6	UK07I1258	17-11-2012	A.B.Gupta	kusumkhara haldwani	856952154	27-01-2016	11-01-2017	0
7	UK07I4521	27-07-2001	S.P.Bisht	new street road roorkee	895658215	27-11-2016	09-02-2017	0
8	UK07K7452	21-04-2006	Y.K.Sharma	near khamba palace haridwar	856593254	29-10-2017	19-07-2016	0

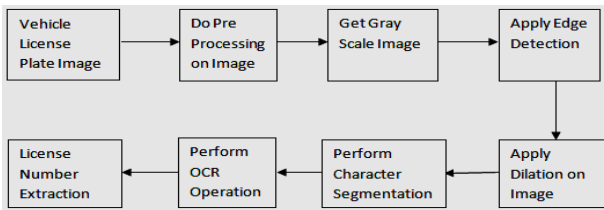
Fig. 1. Screenshot of Dataset used

Note that in the above dataset, the "Stolen\_Status" data can be managed with the help of FIR written in police station and the rest information will be managed from RTO office.

However the most important aspect of this work is clear and accurate vehicle image as much as possible. Since if still there is some issues remain with the image then that will be corrected by using basic digital image processing techniques [10]. So for good vehicle image use good resolution digital camera (suggesting minimum resolution will be 24\*42 pixels). Take care that the number plate must be completely visible while clicking image.

Following are the steps for vehicle number extraction:

- Convert Vehicle Image to Gray Scale Image
- Edge Detection in Resulted Image
- Applying Dilation Operation
- Character Segmentation
- Apply Optical Character Recognition (OCR)



**Fig. 2. Methodology of Vehicle Number Extraction**

In this suggested methodology firstly we have to convert an image into Mat [8] [11] Object since in OpenCV library, which is used here for suggested image processing, there are lots of method associated with it also these are easy to use and much effective also. Many researchers suggested that Digital Image Processing gives a better result if an image is in matrix form, here Mat too store images in matrix form. The steps of extraction of vehicle number discussed here is referred to the paper [6], however for final step we preferred Feature Detection which is one of the OCR technique and much effective then Pattern Recognition used in [6]. Let's discussed more about Feature Detection OCR technique.

In Feature Detection, which is also known as Intelligent Character Recognition (ICR), is much user oriented for spotting characters. It uses set of rules according to each alphabet, each number and each special character. For example, a round shape is 'O'/'o' alphabet, if it is round but the upper portion is not connecting this means 'U'/'u' alphabet and more rules. So using such Features as a rule makes it easy to recognize all characters no matter what is the font style and size. Thus in this technique instead of matching the complete pattern of characters, only individual character's features are detected which makes this technique fast and user-friendly.

This paper suggested to use Feature Detection approach by using Tess4J library. It contains very effective method to apply OCR in vehicle image with the help of Tesseract API. The recognition process in this API is a two pass process. In first pass each character is tried to recognize with the help of an adaptive classifier. However due to different type of font style, some characters are not well recognized in first pass. So in second pass those characters are recognized again by checking alternative hypothesis.

Below is showing this paper's suggested algorithm for Vehicle Number Extraction:

**Algorithm for suggested Vehicle Number Extraction:**

- Step1. Begin
- Step2. Create Mat Object M //convert the image into a mat

- object for enabling java on them*
- Step3. Select Vehicle Image I //variable I is used to store the image
- Step4. Pass  $M \leftarrow I$  //carry out the matrix multiplication
- Step5. Convert to Grey Scale  $\leftarrow RGB$
- Step6. Apply Canny Edge Detection on Grey Scale Image
- Step7. Dilation resultant image from Canny Edge Detection
- Step8.  $M \leftarrow$  subtract Canny from Original
- Step9. Loop till all the pixels of the image are read
  - Assign Image to File F
  - Make Instance of ITesseract // tess4j library is used for OCR
  - Apply Filter  $\leftarrow F$  // to remove noise like lines and left characters only
  - Perform ITesseract to extract number // vehicle number is extracted after this step
- Step10. End for
- Step11. End

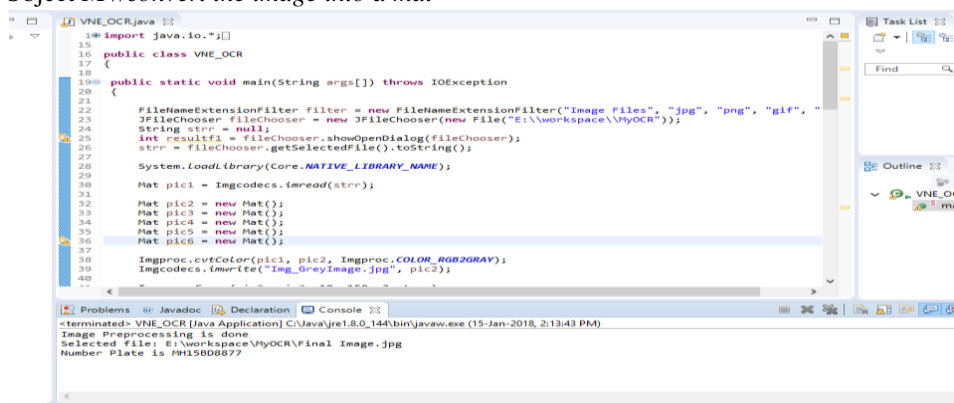
**IV. OBSERVATION OF PROPOSED METHODOLOGY**

For Observation firstly we need to setup computer's environment according to the tools and version given in the Table. I below.

**Table- I: Open Source Tools used for the suggested methodology of this paper**

Tool	Suggested Version
Java	jdk1.8.0_144
Eclipse	Oxygen Release (4.7.0)
OpenCV	opencv-3.3.0-vc14
Tess4J	Tess4J-3.4.2-src
Hadoop	hadoop-2.7.4
HBase	hbase-1.2.6

All these tools are successfully tested and implemented in windows environment (Windows 10) and properly extracted the Vehicle's License Number and storing them into HBase database.

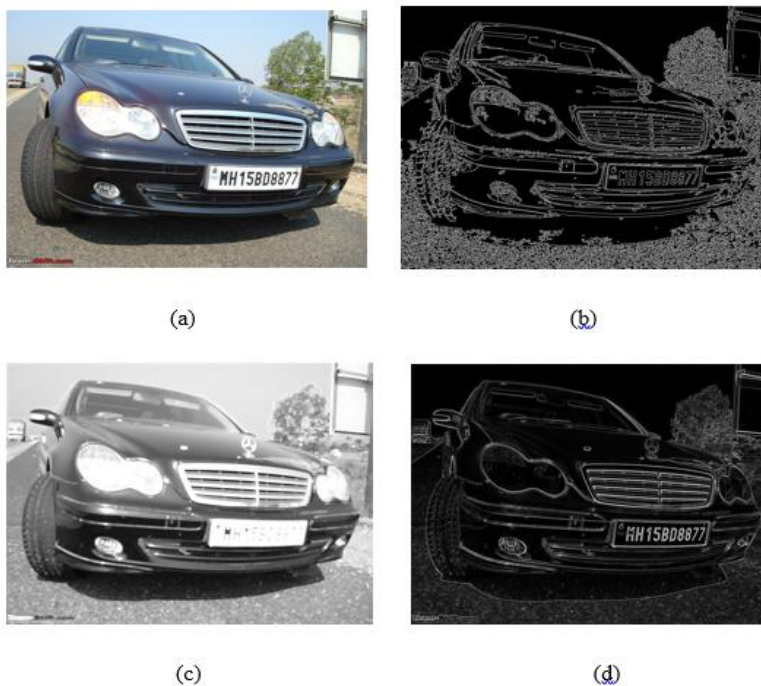


**Fig. 3. Screenshot showing the extracted vehicle number**



## Vehicle Number Management using Hbase

The output of the suggested steps are:



**Fig. 4. (a) Gray Scale Image, (b) Canny Edge Detection, (c) Dilated Image (d) Character Segmentation**

Now once the number is extracted by using the suggested algorithm, as showing in Fig. 3, simultaneously it will be stored in text file also (by using FileOutputStream we can save the resulted characters into any editable text file). Once all records of the dataset is filled in text file, it will be looked

like as given in the Fig. 5 below. We can add any number of vehicles and other information related to it. Note that this data is also acts as a backend data while we perform applications related to this extracted vehicle number, which will be further discussed in this paper.

```

vdetail - Notepad
File Edit Format View Help
UK07G1234:12/07/2001:S.K.Sharma:bhallawala road race course Dehradun:895658452:14/02/2017:12/12/2016:0
UK07G1258:12/08/2002:M.P.Yadav:ladpur Dehradun:854265985:14/03/2017:12/10/2016:1
UK07G5214:30/08/2009:H.K.Verma:new colony abc street haridwar:986584575:24/01/2016:27/01/2017:0
UK07H5468:22/08/2008:C.B.Papnai:uccha pull haldwani:785986231:27/06/2017:17/03/2017:0
UK07I1258:17/11/2012:A.B.Gupta:kusumkhara haldwani:856952154:27/01/2016:11/01/2017:0
UK07I4521:27/07/2001:S.P.Bisht:new street road roorkee:895658215:27/11/2016:09/02/2017:0
UK07K7452:21/04/2006:Y.K.Sharma:near khamba palace haridwar:856593254:29/10/2017:19/07/2016:0
UK07M9568:21/04/2006:Ashish Nainwal:abc road near sbi bank kotdwar:785962354:17/12/2016:24/01/2017:1
UP08G1548:11/11/2009:B.B.Saini:akbar road near abc hotel lucknow:875986524:19/02/2017:21/11/2016:0
UP08G1544:11/10/2011:A.K.Yadav:mathura colony nashvila road sampuranagar:785962532:21/12/2016:31/01/2017:0
UP08G5655:30/10/2008:Jamuna Das:Sarita vihar abc road kanpur:895625425:21/10/2017:21/02/2017:0
UP08D5846:31/12/2001:M.C.Pal:xyz colony sajjahapur biznor:785962545:14/11/2016:09/03/2017:0
UP08B1222:31/12/2003:Y.P.Gangu:near allahabad bank allahabad:986587203:11/12/2016:11/01/2017:0
UP08N5468:21/11/2007:M.P.Verma:patel chowk near baniya sweets Noida:765982542:17/10/2016:05/03/2017:0
UP08M5546:22/10/2001:Ajay pal:abc colony lane 11 shankarpur Allahabad:785965825:12/04/2017:01/02/2017:0
UP08M1154:22/03/2001:Ashish Chandra:new friends colony krishnapur Noida:986582546:12/12/2016:21/01/2017:0
UP06N1004:21/04/1999:Ashish Purohit:old friends colony krishnapur Noida:986588546:12/11/2017:20/09/2017:0
UP06N1044:30/11/1999:Manshi Rawat:near tv tower dehradun:972588546:13/12/2017:10/10/2017:1
UP07N1144:27/01/2007:Meena Rai:tower colony kotdwar:9725877460:11/03/2017:19/11/2017:0
UP07P1148:27/11/2006:Anjali Bisht:ekta vihar kotdwar:9897877460:11/03/2017:19/11/2017:0
UP07P8844:27/12/2007:Chalini Verma:yashwanta colony haridwar:972588860:16/11/2017:10/11/2017:0
    
```

**Fig. 5. Screenshot showing the actual data to be store in HBase**

Algorithm for suggested vehicle data management in HBase database

- Step1. Begin
- Step2. Setup Hadoop
- Step3. Setup HBase
- Step4. Start HBase
- Step5. Create HBase table according to the columns (C) given in Fig. 1 // using queries data will be inserted into HBase
- Step6. F ← read file containing vehicle number details // F is a FileReader class object
- Step7. Loop till end of file // from here data will be inserted into HBase

- Add immutable ← C
- Add column ← C
- Write the data in HBase table
- End loop

- Query to search the vehicle number (N)
- Step8. Flag ← 0
- Step9. Populate list with HBase data
- Step10. Loop till end of list // using queries data will be searched here
- If N found in HBase table then // N is a string having a vehicle number
- Display all result related to N
- Exit loop
- Flag ← 1 // this shows that N is found in the database
- End if
- End loop
- Step11. If Flag is 0 then // this shows that N is not found in the database



Display record not found  
End if  
Step12. End

When we successfully extracted vehicle number and stored it into text file along with other details, as shown in Fig. 1, then it will be stored in HBase using the algorithm discussed above. This proposed algorithm will successfully store all

records available in the text file into HBase and it is capable of storing all real time records also which is the best part of this database. Comparing to other NoSQL database, HBase query response time is very less, which makes it more preferable for real time applications. In windows we can easily view the records stored in the system by using HBase scan query and it will be shown as it is given in below Fig. 6.



Fig. 6. Screenshot showing vehicle records in HBase

Here this database contains three important column i.e. “Last Road Tax Deposit Date”, Last Insurance Date” and “Stolen Status of the Vehicle”. As according to the suggested algorithm (Algorithm for suggested vehicle data management in HBase database) in this paper, when one enter vehicle number to search it in database, it will show all records present in the database. For example when we search vehicle number “UP07Q1194” it will shown as:

```
Address: janta colony kotdwar
Contact: 9720877411
Last_Insurance: 19/01/2016
Last_RTax_Dep: 11/09/2016
Name: P.K.Kandpal
Regist_Date: 29/11/2009
Stolen_Status: 1
Veh_Num: UP07Q1194
```

Now let today’s date is 20/09/2019, so it clears three things. First one is that insurance is not renewed by this vehicle, second is that this vehicle has not paid the road tax (assume it as yearly basis) and finally an important thing is that the stolen status of this vehicle is 1 that means this is a stolen vehicle, appropriate actions will be taken against the person who brings this vehicle. What this paper wants to describe here is that once a vehicle comes under surveillance, if it’s a new vehicle we add it into HBase database or if it is presented in our database we can check and analyse the details as discussed above.

This result can be used in many applications in which vehicle number is used as an object. Like it can be used by traffic police to identify stolen vehicles as well as those vehicle whose vehicle insurance is due and who has not deposited the road tax. This is also used in RTO office to figure out those vehicles whose registration of vehicles are expires (assume it as 10 years basis). These can be apply in real time scenario also and also the whole methodology suggested in this paper not only effectively extracting the vehicle number but also managing them in HBase database, no matter either it’s a data only or big data, also this paper

provides an user oriented environment to apply this methodology in simple as well as real time applications like detecting stolen vehicles, detecting road or vehicle tax and more.

V. RESULTS

The proposed methodology is giving satisfactory result with different type of images along with different image size. For comparison the paper opted the ANPR which is an automatic vehicle identification system.

Automatic Number Plate Recognition (ANPR) is an image processing technology which uses color image of number plate then it automatically sense the number plate image and in output it shows the number printed on the vehicle image.

However the framework discussed in this paper is application based and also it is not only far better than ANPR in terms of vehicle number extraction but also it provides a user oriented environment in which we can perform the above discussed applications. The proposed framework is compared with the ANPR and it is observed that the prior giving better result than the later. Robustness of the proposed algorithm is checked on the basis of precision and recall for different query image. Precision and recall are calculated using:

$$precision = \frac{\text{number of relevant result}}{\text{total number of retrieved resu}}$$

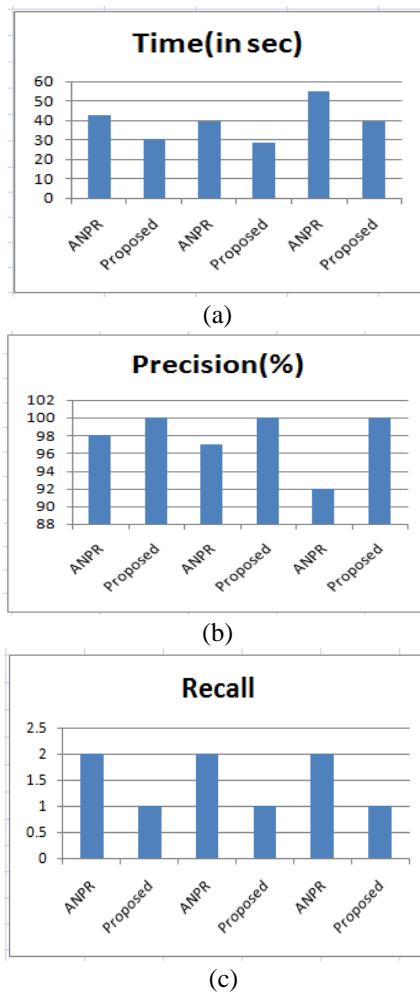
$$recall = \frac{\text{number of relevant result}}{\text{total number of expected results}}$$

Table- II: Results Obtained of Time, Precision and Recall

Technique	Image size	Type	T (in sec)	P (%)	R
ANPR	128 X 128	jpg	42.54	98	2
Proposed	128 X 128	jpg	30.6288	100	1
ANPR	128 X 128	bmp	39.20	97	2
Proposed	128 X 128	bmp	28.2240	100	1

ANPR	256 X 256	jpg	55.054	92	2
Proposed	256 X 256	jpg	39.6388	100	1

(Here T=Time, P=Precision, R=Recall)



**Fig. 7. Graphs showing comparison between ANPR and Proposed Methodology in this paper on the basis of Time, Precision and Recall respectively**

### VI. CONCLUSION

The Discussed Methodology in this paper of detecting License Number of Vehicles and detecting them from HBase database is precisely tested into Windows environment by implementing it in Eclipse using Java as programming language. All experiments are done using real-time Vehicle Images. All suggested steps and suggested algorithms are attentively implemented over the Vehicle Images and can be implemented on live streaming also. Whether the image is taken from the front side or back side or even if image quality is poor it will still be able to extract the number from the image on applying the above suggested algorithm. OpenCV Library and OCR API (Tesseract) with filtering the output using simple text mining steps make it possible to achieve good results. Apart from the extraction of vehicle number, its storage in HBase database makes it highly applicable since HBase gives best analytic results under real time processing of vehicle traffic. So once the License Numbers are stored in the database we can use this data in many applications like Vehicle Parking Access Control, Automated Toll Tax Calculator, Detecting Stolen Vehicles and more. For future,

the authors will try to enhance the functionality of this suggested application and also incorporate some security mechanism over it to make this secure since database has personal information being extracted from the vehicle numbers. Security using encryption can be implemented in later stages.

### REFERENCES

1. K.K.Kim et al., "Learning-Based Approach For License Plate Recognition", in INNSP X, vol. 2, Sydney, NSW, Australia, 2000, pp. 614 - 623.
2. Shyang-Lih et al., "Automatic License Plate Recognition", in TITS, vol. 5, IEEE, 2004, pp. 42-53.
3. Rana Gill, Navneet Kaur, "Indian Vehicle Number Plate Detection by Image Processing Technique in MATLAB", in IJCST, vol. 3, 2012.
4. Priyanka Prabhakar, Anupama P, Resmi S R, "Automatic Vehicle Number Plate Detection And Recognition", in ICCICCT, Kanyakumari, India, 2014, pp. 185-190.
5. V Lakshmi Priya, K. Perumal : "Detecting the Car Number Plate Using Segmentation", in IJECS, vol. 3, 2014, pp. 8823-8829.
6. Pratiksha Jain, Neha Chopra, Vaishali Gupta, "Automatic License Plate Recognition using OpenCV", in ijcat, vol. 3, 2014, pp. 756-761.
7. Anumol Sasi, Swapnil Sharma, Alice N. Cheeran, "Automatic car number plate recognition", in ICIECS, Coimbatore, India , 2017, pp. 1-6.
8. Yingjie Xia et al., "Big Traffic Data Processing Framework for Intelligent Monitoring and Recording Systems", 2015,. doi: 10.1016/j.neucom. 2015.07.140
9. P.V. Suryanarayana et al., "A Morphology Based Approach for Car License Plate Extraction", in Annual IEEE India Conference, Chennai, India, 2005, pp. 24-27.
10. Bhonsale Tejas et al., "Number Plate Recognition and Document Verification using Feature Extraction OCR Algorithm", in ICICCS, Madurai, India, pp. 1317-1320.
11. Ainhoa Azqueta-Alz'uaz et al., "Massive Data Load on Distributed Database Systems over HBase", in CCGRID, Madrid, Spain, 2017, pp. 776-779.
12. Xiaomeng Zhao et al., "Metadata Extraction and Correction for Large-Scale Traffic Surveillance Videos", in IEEE International Conference on Big Data, Washington, DC, USA, 2014, pp. 412-420.
13. Dandan Shen et al., "A Nearby Vehicle Search Algorithm Based on HBase Spatial Index", in 12th WISA, Jinan, China, 2015, pp. 71-74.
14. Dawen Xia et al., " A Map Reduce-Based Nearest Neighbor Approach for Big-Data-Driven Traffic Flow Prediction", in IEEE Access, vol. 4, IEEE, 2016, pp. 2920-2934.
15. Jianhua Xie, Jiahui Luo, "Construction for the city taxi trajectory data analysis system by Hadoop platform", in 2nd ICBDA, Beijing, China, 2017, pp. 527-531.
16. Li Zhu et al., "Big Data Analytics in Intelligent Transportation Systems: A Survey", in IEEE Transactions on Intelligent Transportation Systems, vol. 20, IEEE, 2018, pp. 383-398.

### AUTHORS PROFILE



**Mr. Chetan Pandey** is an Assistant Professor in the School of Computing, Graphic Era Hill University, Dehradun (UK). He has received Master in Technology in CSE from Graphic Era Deemed to be University, Dehradun (UK). His research interests are Big Data Processing and Machine Learning.



**Mr. Amit Juyal** is an Assistant Professor in the School of Computing, Graphic Era Hill University, Dehradun (UK). He has received Master in Technology in CSE from Graphic Era Deemed to be University, Dehradun (UK). His research interests are Big Data and Machine Learning.



**Ms. Poonam Verma** is an Assistant Professor in the School of Computing, Graphic Era Hill University, Dehradun (UK). She has received Master in Technology in CSE. She worked in the area of the Machine learning algorithms with python programming. specifically worked with the supervised learning models.



**Dr. Kamred Udhham Singh** is an Assistant Professor in the School of Computing, Graphic Era Hill University, Dehradun (UK). He has received Ph.D. degree from Banaras Hindu University, Varanasi. His research interests are Image Steganography and Image Watermarking.