

License Plate Recognition for Traffic Management

Astha Singh¹, Pawan Singh²

Amity School of Engineering and technology, Amity University, Lucknow, India^{1,2}
singhastha2809@gmail.com¹, pawansingh51279@gmail.com²

How to cite this paper: A. Singh and P. Singh (2021) License Plate Recognition for Traffic Management. *Journal of Management and Service Science*, 1(2), 1, pp. 1-14.

<https://doi.org/10.54060/JMSS/001.02.001>

Received: 15/05/2021

Accepted: 26/05/2021

Published: 27/05/2021

Copyright © 2021 The Author(s).

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The objective of this report is to present an overview of the project, license plate recognition. The system is to basically detect a vehicle using the information of the license plate in order to use information at various valid sites. This tool can work as a part of other big projects in the industry for security purposes as well as for analysis purposes. There is a detailed insight of the project several different chapters throughout the report. This project is based on detection and recognition algorithms; using several libraries of python to work on images and videos and thereafter using processed image to further train and test a model using machine learning algorithm such that recognition is done with higher accuracy. The beginning outlines the introduction to the topic and importance in the real world by highlighting some other applications using similar approach. Later highlights the technology and skills in use in order to completely deploy the idea. Further there is explanation to the feasibility study as well as the requirement specification while system design involves around the basic designing of several modules which would integrate to work as the whole system and test cases. Implementation and testing approach is being discussed and thereby light been thrown on the results and conclusion thus bringing attention to system's limitation and its future scope.

Keywords

Machine Learning, Detection, Segmentation, Prediction, Training Model, Recognition

1. Introduction

1.1 Background

License Plate Recognition is one of the solutions well it comes to traffic/ vehicle management systems. Since with time vehicle crowd on road is increasing day by day and the need for a system to work efficiently without manual help has become a need. Intelligent transportation system can comprise of small systems making use of artificial intelligence, internet of things, etc. and other newest technologies in order to manage vehicles at different places such as, parking lots, toll plaza, security purposes, for easy traffic rules management, etc. Since urbanization has seen its growth in the past few years with which population as well as utilities have increased to its zenith, it has become a goal to invest in technologies to modify the style of living to bring ease to human lives. [4] As we talk about one of the solutions to manage vehicles that is License Plate Detector which can work in tracking down the vehicle by tracing down the characters of the license plate. The need is to identify single out of a whole

lot of vehicles. As a matter of fact every country has its own pattern and style prescribed for the number plates of their own country. There should ideally be particular and uniquely trained systems for different countries in order to work accurately and efficiently for that country. [2] For example, Chinese number plates have alphanumeric characters along with some Chinese symbol unlike an Indian license plate which is just formed by mix of alphanumeric characters. [7] Also, it is evident that different countries have different colour scheme for the number plate of the vehicles for their country.



Figure 1. License Plate Detector deployed at a parking lot

In figure 1. We can observe that a system of automatically detecting license plate has been installed in a parking lot in order to efficiently working with the parking spaces, charges, etc. This can be achieved by object detection and object recognition which will be further discussed in the upcoming chapters. To make this possible and deploy it as various sites along with the usage of some cameras, recorders can be a good idea to bring it into working.

1.2 Objective

The main objective is to design an efficient and automatic system to identify vehicles by using information of the license plates. For an effective ALPR system, aiming at improving the recognition algorithms efficiency for increasing the accuracy of the plate detection and character recognition. The system can then be used at various sites of application in order to trace the required vehicles.

1.3 Purpose, Scope and Applicability

1.3.1 Purpose

The objective of choosing this project is to learn and implement technology in order to serve real time needs. As theoretically there are many inputs to this kind of system, there are number of papers and citations that talk about how we can detect plate and recognize its characters in various possible as well as hypothetical methods to implement the same. [2] There can be many machine learning algorithms that can be brought into picture as well as there can be use of neural networks to recognize the results. The project thus built taking all aspects into consideration has its sole purpose to work on the accuracy of the detection such that the system can actually be deployed and brought into use.

1.3.2 Scope

The project holds its importance when it comes to an efficient on road system. The method involves capturing the images and videos and working on them in order to extract the plate and hence the characters on the plate. The steps involved would be

capturing the video/ image, image processing, detection of the number plate, character segmentation (to read each character clearly and separately) and character recognition finally providing the required result. The limitations of the project is currently it cannot work with multiple vehicles in the picture, that area is still what needs to be worked upon. An input with right amount of light and colour would give more accurate result. A dark image would limit the functionality of the system as it is based on detecting the plate using gray scale image and thus black and white colour gradients. The assumptions made are that the input feed has balanced colours, since Indian number plates are a combination of black and white colour, working keeping that feature in mind would help extraction of the plate easy and accurate.

1.3.3 Applicability

A system that has an ability to detect, locate and identify number plates out of the camera feed can be used at various places to monitor transportation system with least manual efforts. There are increasing number of road accidents so not only at ideal places but also at crime scenes, this kind of system can be very useful for security management as well as law enforcement. A manual identification is prone to errors but an accurate system would prove to be time saving as well as it would be less prone to commit mistakes in tracing the license plate characters. The direct field of application can be providing the system with inputs such as image and video feeds such that it could provide the resulting number plate characters. [7] The indirect and long term future scope could be syncing a CCTV camera or a drone deployed with an image capturing device such that it could feed the system with real time inputs and get the effective result where the system is capable of identifying the vehicles using their license plates. Those CCTV cameras can be installed at several places like toll plaza, parking areas, highways, etc. wherever needed and finds its use. [9]

1.4 Achievements

The weeks invested in this project has definitely boosted my confidence in working with different data, algorithms and thriving to achieve accuracy, finding out the bugs that are compromising on the system's efficiency. Trying different methods stated by different researchers and concluding how hypothetical or realistic their approaches towards the problem was has given me potential to try all possible methods instead of depending on others conclusion. There were many researches of using different algorithms and achieving minimum time complexity which on trying proved to give a loophole at some point of time where training data made the model resulted in over-fitting or under-fitting. [5] Working on detection by using several python libraries (further discussed in Chapter 2) familiarized me with image manipulation and simultaneously preprocessing. There were many online available resources and documentation throwing light on the same proving to be helpful and thus implementation for the same was made smooth. While training the model, finding of an appropriate dataset and also managing to make a new self-made dataset due to lack of enough data was a challenge but since it was an important step, accuracy could not be compromised on anything which is possible to do. Finding the correct algorithm to deal with recognition portion of the system was also a learning curve as it brought forth the pros and cons associated with different algorithms and thus helped me in summarizing everything down to one most suitable algorithm (further discussed in Chapter 2). Analyzing at each step what was the requirement and how it can be fulfilled was filled with so many hits and trials that it contributed in several prototypes and the best suited as well as best performing prototype was polished and worked upon to give the final best system for an automatic license plate recognition system. The goal was to achieve accuracy at its best which with some assumptions keeping in mind it is a machine based model comes some limitations, an amazing result was thus achieved giving a significant boost to my learning curve.

2. Survey of Technologies

When a human looks at an image or a video stream he/she is able to recognize the objects separately as well as they can spot them easily with identifying almost all the visual details. If a similar thing can be obtained by a computer we can possibly make it retain some wonderful human feature which is a natural one for us but might sound difficult for a machine. [2] We can find and identify objects as well as name them just by seeing it, a machine needs to be programmed or taught to get that capability, this could be termed as **object recognition** [6] in plain terms which is a computer vision technique working on the images using some machine learning and deep learning algorithms. This ability is renowned for its massive applications in driverless cars identifying signboards or other objects which is a symbol for these for when to apply breaks and when to drive. Object recognition comes next to object detection.

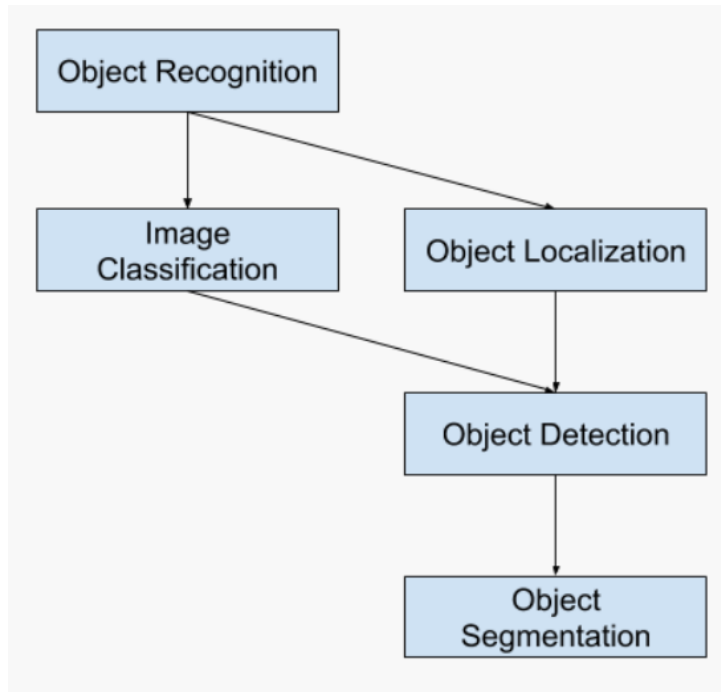


Figure 2. Overview of Object Recognition Computer Vision Tasks

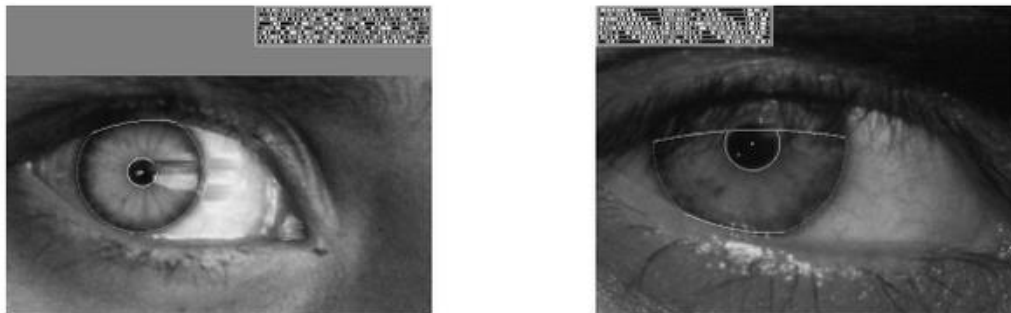
2.1 Real-Time Applications of Object Detection and Object Recognition

2.1.1 Optical Character Recognition: There are a few industries which work with texts, characters and for such industries different combination of characters have special and distinct meaning. For example, detecting the character on the license plate.



Figure 3. Optical Character Recognition

- 2.1.2 Self-driving Cars:** This is a classic example and also amongst the major field involving innovation and technologies. Here detection of object helps in identifying the actions in sequence for a smooth rendition of the driverless cars.
- 2.1.3 Tracking objects:** In several sports as well as industries where little movements make a difference, object tracking is important such as medical field, a cricket match, a security system tracking human movements, etc.
- 2.1.4 Face Recognition:** Detecting faces are widely used in security, social networking sites for tagging purposes, etc. Also we must not forget the idea of frequently using face unlock in our smart phones.
- 2.1.5 Human features recognition:** Recognizing iris for security and using biometrics for different purposes Iris recognition is considered widely for high level security and scanning purpose, highly recommended for accuracy for biometric.

**Figure 4.** Human feature recognition

- 2.1.6 Object extraction from image or video:** Classification, sequencing actions, Segmentation for meaningful insights and usage of obtained image accordingly as per the requirement is one of the major usefulness here.
- 2.1.7 QR Code scanning:** We use this in our daily lives making payments, scanning codes for other useful information, etc. has made us realize the importance of precision while object detection and recognition.
- 2.1.8 Activity Recognition:** This is majorly used for knowing the sequence of events to be followed or completing the missing frames while rendering a video with missing frames, can be utilized to predict the movements to make system secure or for making systems ready for the action in queue.
- 2.1.9 Medical Imaging:** To identify several unwanted presence of any object for giving meaningful insights for better diagnostics, surveillance, therapy planning and other crucial tasks.

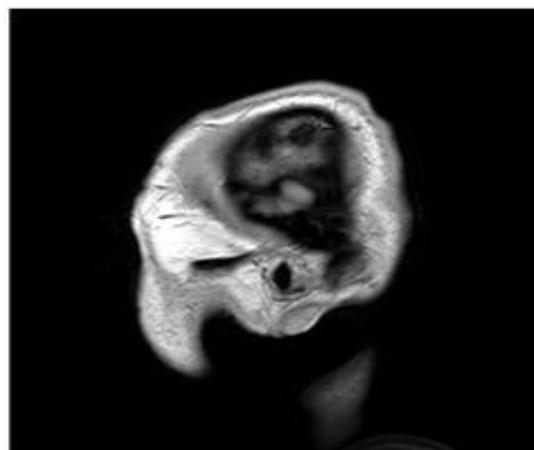
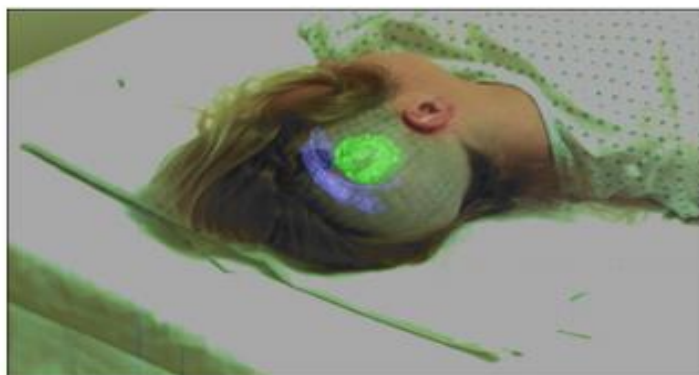


Figure 5. Medical Imaging

- 2.1.10 Using image in search engines:** The newest feature allowing us to search the related results using image search as an option is really helpful.
- 2.1.11 Scanning OMR sheets:** In examinations held at all levels, the use of OMR sheets is done to implement efficient checking, this is also an amazing example where object recognition has a role to play.

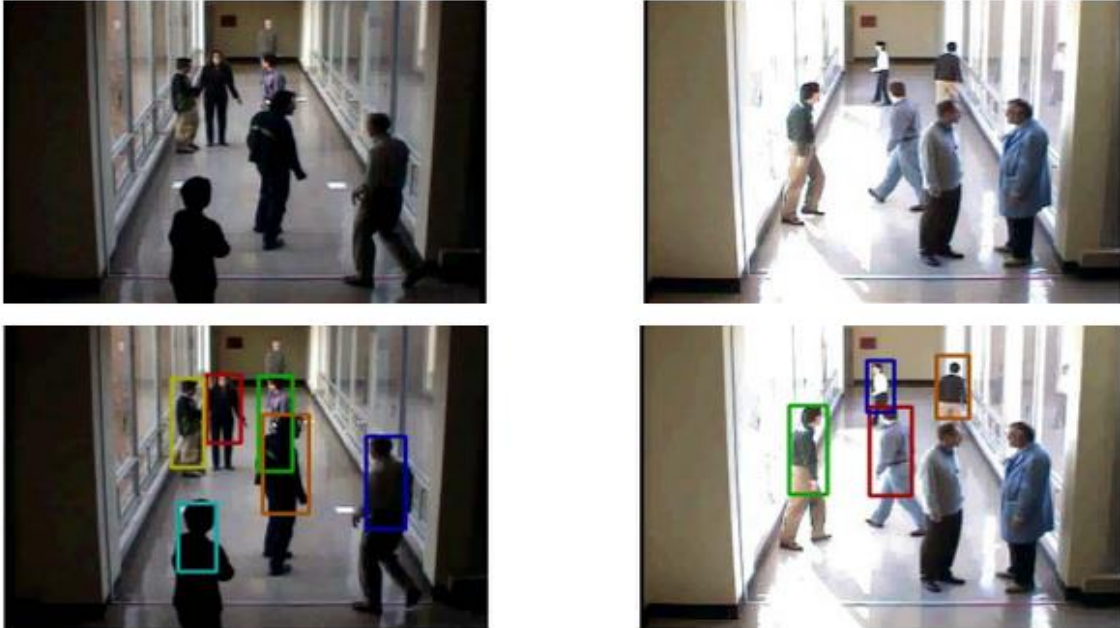


Figure 6. Human Recognition in real-time

There are end numbers of applications, few being stated above. This signifies how important can computer vision be making our lives easy reducing human efforts in all possible ways [10-13].

2.2 Implementation of Object Detection

There are several ways to think about how to start with object detection.

- 2.2.1 YOLO “You Only Look Once”** is an important member of the CNN (Convolutional Neural Networks) family which is has popular models for its performance in real-time object detection. YOLOv3 for keras library is the best for object detection. It is an open source library.
- 2.2.2 Object Detection API of tensorflow** is another library popular for Object detection, it is an opensource framework. It provides easy implementation of object detection as a whole.
- 2.2.3 HOG (Histogram of orientated gradients)** is used to detect objects detecting descriptive features in an image.
- 2.2.4 SSD i.e. Single Shot Detector** is used for detecting objects in an image using a single (as the name suggests) deep neural network, that is why it is called “single shot”.
- 2.2.5 Using python script** simply using libraries like OpenCV – for working with images , scipy –for working with object localization and dimension analysis, imutils – for working with bounding boxes, numpy – for working with multi-dimensional array functionalities, etc. [8]

There are several other methods to do the same.

3. Implementation and Testing

3.1 Implementation Approaches

The steps involved in implementation of the project are as shown in the below diagram.

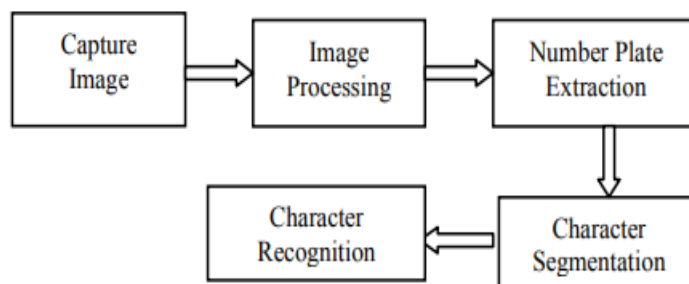


Figure 7. Flowchart for steps involved

The further implementation will be explained in the following section. The capturing of image or video feed will work as input to the next step that is image processing which is important for further image manipulation to reach the goal state where an image of number plate will be obtained. The next step will only be possible if correct extraction is done. A model will be trained to recognize characters further. [2] Then the segmented characters will be sent to the model to individually give the result of the number plate.

3.2 Coding Details and Code Efficiency

First step involves **detection or locating the license plate**. Libraries or modules like, scikit-learn, imutils, skimage, OpenCv, matplotlib, os, shutil will be required. Eventually we will walk through the major functionalities of these modules that contribute to the code widely and have made this step possible. From skimage.io we import imread plugin which is useful for image reading and writing, from skimage.filters we import threshold_otsu which works on the Ostu's Method to return threshold value for the image or histogram input. [1] From skimage we import measure module has many functionalities like measuring region's properties; like from skimage.measure import regionprops is used to label the region of the image along with the measure properties. [3] The OS module of python is used to manipulate files and provides a way for using OS functions efficiently. The shutil module of python is also used for manipulating files or its collection that is directories. This module is similar to the OS module of python.

We read a path which stores the video data locally on the device and check for the directory named output, if it exists locally, we delete it and create a new directory with the name "output". Then we can use the VideoCapture function from OpenCv module in order to read the frames of the video data which was read using the path in gray scale. Using functions and loops we are reading all the frames and saving it in the created directory which we created. The frames that are being read can be stopped by using the key q or it will automatically end at the time lapse provided in the waitKey function passed as the attribute to the function. Once the video feed is done we should release all the resources in use and thereby destroy the window which is being displayed. [8]

The next step would be reading the particularly second last frame and saving the image as a gray scaled one in order to perform further image processing to extract the plate. Then we may use rotate function available in imutils module. Thus, conversion of the particular frame into gray scale image and then to a binary image is the aim. Since the captured frame which we will be working on is a gray scale image its pixel falls in the range [0, 1] and making it 255 times by multiplying it can now

falls in the range [0, 255]. Using `thresholds_otsu` method available in `scikit image` module we are extracting a binary image of the same after availing the threshold value and thus converting the gray scale image into binary image for further object detection. [1] We are displaying the two images side by side to know the difference. The threshold value actually represents the value of upper limit for which all the pixel intensities are considered to form the foreground image. Next step is to find out the connected regions of the obtained image such that we can find the objects we are looking for. We use the `measure` module of `python` in order to do the same. Connected regions corresponds to making group of connected pixels which are called connected if they are neighbours and have same value. [9]

We encounter greater problems when trying to count the number of turkeys in

Labeling the thresholded image



yields

as a graylevel or



as a color labeled version. Although we managed to assign one connected component to each turkey, the number of components (196) does not correspond to the number of turkeys.

Figure 8. Gray scaled Image

The next step would be to identify what could be the maximum and minimum dimensions of the number plate including height and width. To measure the characteristics of the previously obtained regions of the labeled image a loop can run so as to trace all the regions with valid properties that stand valid for the case of a number plate. As soon as we are available with the region properties that matches are set requirements we extract that much region by applying bounding box around the available region. Since we have defined two possible properties of an ideal bounding box to trace the number plate, we will perform the task twice if in first attempt the match is not found. Finally we may obtain the area required as per the specifics we have fed the functions with. [6]

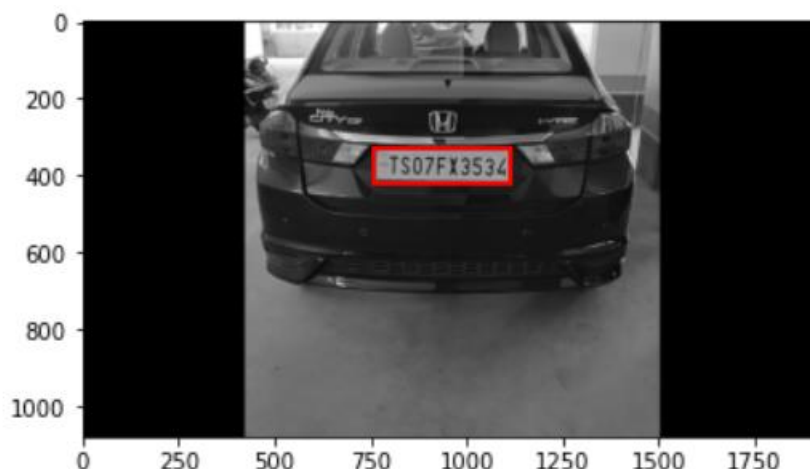


Figure 9. Detection of number plate

Second step involves **segmentation of the characters on the number plate**. Libraries or modules like, scikit-image, numpy, skimage, matplotlib will be required. Eventually we will walk through the major functionalities of these modules that contribute to the code widely and have made this step possible. From numpy module we will use invert function in order to invert the pixel values such that the colour inversion is done. Now the first element of the list containing plate like objects will be manipulated. The number plate is originally having white background and black inked characters onto it. On applying inversions the characters can be visualized as white on a black background. The manipulation is done on the array (since all gray scaled images can be viewed as a 2-D array). Next step is to find out the connected regions of the obtained plate image such that we can find the characters individually we are looking for. We use the measure module of python in order to do the same.



Figure 10. Extracted plate

Connected regions as already discussed while extracting plate like objects from the image is similarly done here for character extraction as our aim is to individually get them for recognition further. The strategy is same as done for plate extraction; we will work on several assumptions like assuming character dimensions are certain percentage of the plate. [6] This would include identifying what could be the maximum and minimum dimensions of the characters in the number plate including height and width. To measure the characteristics of the previously obtained regions of the labeled image a loop can run so as to trace all the regions with valid properties that stand valid for the case of a character. As soon as we are available with the region properties that matches are set requirements we extract that much region by applying bounding box around the available region. We will resize the output image in order to increase visibility. The sample segmentation result obtained is as follows.



Figure 11. Segmented characters

Once the segmentation of the available characters recognized by using several modules and functionalities is done we move ahead to train a model to predict the characters efficiently.

Third step involves **training of a model in order to predict the characters**. Libraries or modules like scikit image, scikit learn numpy, skimage, os will be required. Eventually we will walk through the major functionalities of these modules that contribute to the code widely and have made this step possible. From skimage.io we import resize plugin which is useful for image reading and writing, from skimage.filters we import threshold_otsu which works on the Ostu's Method [1] to return threshold value for the image or histogram input. From sklearn.svm we now import SVC and sklearn.model_selection we import cross_val_score. Now these libraries and modules will help us proceed in this step of the project. [4]

Initially a list for all alphanumeric characters is defined. We have already downloaded a dataset for characters from kaggle in order to train a model for character recognition. This dataset is preprocessed and cleaned and thus has been used, it is a dataset for character images as that on the number plate. Now the first task is reading the character images for each and every character. As we know that whenever we are dealing with images we need to convert it into its binary form in order to manipulate it further for significant utilities. So our goal is to read and convert all the images in binary format. We are using threshold_otsu [3] function that which has already been used in plate detection is similarly used here too. Since now we are into training a model there are many methods to opt for an algorithm or neural network but here we have used support vector machine as it gave very precise results and was time efficient. [8] The goal now would be converting 2-D array representing images to 1-D array because of the fact that an algorithm in machine learning always takes simplified arguments in order to give result, thus, the 20 x 20 image will have to be converted to 1 x 400 that means for each pixel feature we will be having four hundred features defining it individually. Once the reading and initial manipulation is done we can proceed further in training of the model for future use. [9]

As we know that a support vector machine can have kernel values as either of the three: linear; polynomial or rbf, here we are initializing it to linear and in order to see how sure was the model of its prediction we have initialized probability value as true. An object for SVC is thereby created and for checking accuracy of the model we are using cross-validation method where we have done it for four folds (that is the complete dataset is partitioned into four and one by one for each of the four loops $\frac{1}{4}$ part is used for testing the model after it has been trained on $\frac{3}{4}$ part of the dataset). This object of SVC (a classification algorithm) has been fitted and transformed for the previously read 1-D image data and its corresponding target data. This model is finally saved using the pickle module of python that deals with loading and saving of the models. [6]

```
Cross Validation Result for 4 -fold
[ 96.47058824 100.          95.29411765  98.82352941]
```

Fourth step involves **prediction of the segmented characters**. This is the last step of the project and it involved prediction on the basis of the model saved in the third step. Libraries or modules like pickle will be required. Since all process has been initially

done we are just required to load the model and work with the input feed. Eventually we will walk through the major functionalities of these modules that contribute to the code widely and have made this step possible. [2] We will load the saved model which is a classifier and will classify the input character in either of the classes and hence provide the output. We are using this model for each and every segmented character that has been recognized in step two of the process. We are appending each segmented character's prediction result and finally displaying it. Lastly in case there is some sequencing issue in the result we are using column list and sorting it. [3] Finally, we are tracing the correct index of the predicted output in order to display the result for which we had aimed at the beginning.

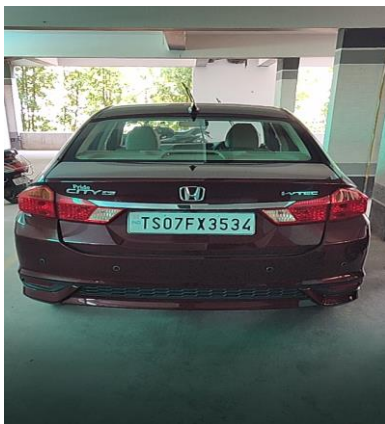
```
Predicted license plate
TS07FX3543
License plate
TS07FX3543
```

4. Results and Discussion

We will view a few sample cases against which system will be tested and thus analyzed. As the model can be tested at each stage or step let us take a sample in order to get results at each level.

4.1 Sample Test Reports

Sample test case 1: input

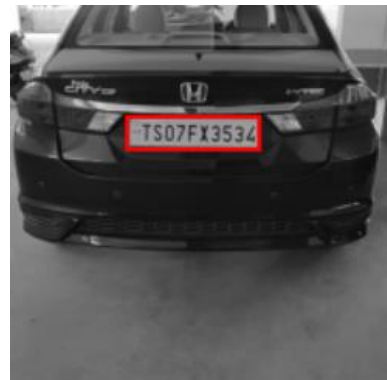


Character segmentation



Preview on frontend

Detection of number plate



Prediction of the characters

```
License plate
TS07FX3534
```



Figure 12. Sample Test Case 1

Result: The result was achieved with 100% accuracy at each level and the plate was detected, characters were segmented and were predicted accurately at each level of the system. The model is perfectly classifying each character with much more efficiency than that which was observed when work was done using CNN.

Sample test case 1: input



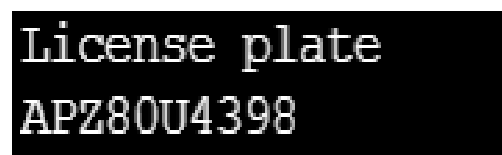
Detection of number plate



Character segmentation



Prediction of the characters



Preview on frontend

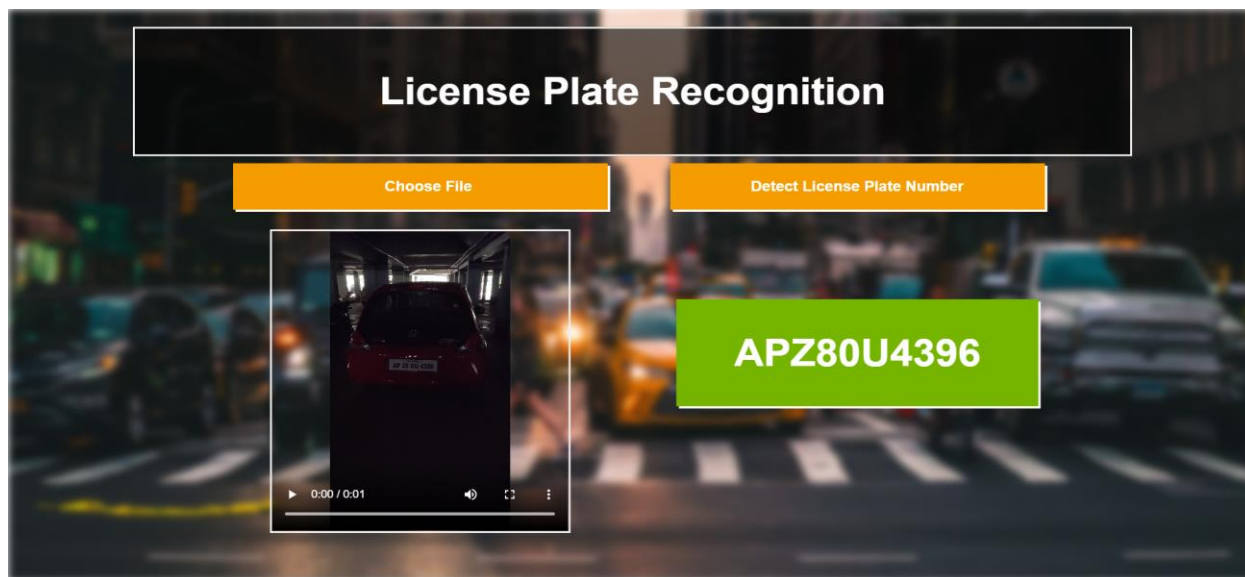


Figure 13. Sample Test Case 2

Result: The result obtained comes with 96% accuracy. The steps like detection of the plate, segmentation of the characters has been done 100% accurately even in low light but the character 2 in actual plate is being read as Z and the character D in actual plate has been classified as O by the model. Hence we see that there are some points at which error may occur in classification. On the other hand, it is evident that the detection/ localization and segmentation has been perfectly done even for low light.

5. Conclusion

The aim to build this project was to efficiently reduce time and increase accuracy. There are a number of methods already defined for license plate detection and recognition in the market but none has been brought to real life practice, if the computation is less and the performance is not compromised this can be done. In this model that has been discussed majorly focused on the power of the libraries available that can filter out the background and extract the foreground as the plate itself and then further work on several mathematical formulae to segment the characters and thus work with prediction. There are many techniques for both the steps detection as well as recognition and implementing the best one could offer a decent decrease in human efforts on sites where the application of this project is in use. [3] We have seen how noise is removed and how efficiently the segmentation has brought forth the further steps' success. As the steps are interlinked like a chain any one of the bead being broken can lead to failure of entire system hence we can note that we have efficiently and smoothly made system work at each levels at its best. The proposed project can be deployed and can be brought in use. There had been several attempts made to make a system that is capable of performing under several climatic conditions that may affect the image in use. Need for upgrades to match the feed that is being processed by the camera that could match the speedy vehicles and capture an informative image has increased, if that problem is dealt with in future, the system proposed could work on it as deployment is successful if it gives information. With the increase in transportation and technology all the thoughtful logical systems can come into picture with it. LPR is one of the very needful systems in near future.

Acknowledgement

I would like to thank my university for giving me this golden opportunity to research on such an interesting topic. I would also like to thank Dr. Anil Kumar, Assistant Pro Vice Chancellor & Director, ASET and Dr. Deepak Arora, Professor & Head, Dept of CSE & IT, ASET for encouraging students to indulge in valuable research activities to enhance their technical skills. Next, I want to thank my faculty guide Dr. Pawan Singh for guiding me throughout the course of this project. I would also like to thank the internet, books and the institute for the knowledge I acquired with their help.

References

- [1] H. Karwal and A. Girdhar, "Vehicle Number Plate Detection System for Indian Vehicles," 2015 IEEE International Conference on Computational Intelligence & Communication Technology, 2015, pp. 8-12, doi: 10.1109/CICT.2015.13.
- [2] S. Saraswathi, R. Subban, T. Shanmugasundari and S. Manogari, "Research on License Plate Recognition Using Digital Image Processing," 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2017, pp. 1-6, doi: 10.1109/ICCIC.2017.8524147.
- [3] Q. Wang, "License plate recognition via convolutional neural networks," 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2017, pp. 926-929, doi: 10.1109/ICSESS.2017.8343061
- [4] S. N. H. S. Abdullah, K. Omar, S. Sahran and M. Khalid, "License plate recognition based on support vector machine," 2009 International Conference on Electrical Engineering and Informatics, 2009, pp. 78-82, doi: 10.1109/ICEEI.2009.5254811.
- [5] Boutaba, R., Salahuddin, M.A., Limam, N. *et al.* A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *J Internet Serv Appl* 9, 16 (2018).
- [6] A.Singh and P.Singh (2021) A Comprehensive Survey on Machine Learning. *Journal of Management and Service Science*, 1(1), 3, pp. 1-17
- [7] Z. Huang and L. Hou, "Chinese License Plate Detection Based on Deep Neural Network," 2018 International Conference on Control and Robots (ICCR), 2018, pp. 84-88, doi: 10.1109/ICCR.2018.8534484.
- [8] L. Li, Y. Wu, Y. Ou, Q. Li, Y. Zhou and D. Chen, "Research on machine learning algorithms and feature extraction for time series," 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017, pp. 1-5, doi: 10.1109/PIMRC.2017.8292668.
- [9] L. Zheng and X. He, "Number Plate Recognition Based on Support Vector Machines," 2006 IEEE International Conference on Video and Signal Based Surveillance, 2006, pp. 13-13, doi: 10.1109/AVSS.2006.82.H.Obeid and R. Zantout, "Line Processing: An Approach to ALPR Character Recognition", 2007 ACS/IEEE International Conference on Computer Systems and Applications, Amman, Jordan, May 13-16-2007
- [10] O. Obulesu, M. Mahendra and M. ThrilokReddy, "Machine Learning Techniques and Tools: A Survey," 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), 2018, pp. 605-611, doi: 10.1109/ICIRCA.2018.8597302.
- [11] A. Singh, P. Singh (2020) Image Classification: A Survey. *Journal of Informatics Electrical and Electronics Engineering*, 1(2), 2, 1-9.
- [12] P. Singhal, P. Singhand A. Vidyarthi (2020) Interpretation and localization of Thorax diseases using DCNN in Chest X-Ray. *Journal of Informatics Electrical and Electronics Engineering*, 1(1), 1, 1-7.
- [13] A. Singh and P. Singh (2021) A Comprehensive Survey on Machine Learning. *Journal of Management and Service Science*, 1(1), 3, pp. 1-17.