

Automatic License Plate Detection and Recognition (ALPDR)

Mohd Amaan Abbasi, Ms. Anam Saiyeda
{ amaanabbasi99@gmail.com, anamsaiyeda@jamiahamdard.ac.in }

Jamia Hamdard, Hamdard Nagar, New Delhi-110062, India

Abstract

This work attempts to solve the problem of license plate detection and recognition of its characters. It also aims to improve various parts of the detection and recognition pipeline and proposing new state of the art techniques to make the pipeline robust. The cases with naïve approach are discussed, which could be a better option than a deep learning technique. The goal is to make the model accurate and fast, so as it can produce real-time results. The model is kept light weighted and formatted in such a way that it is easily deployable to different platforms. This paper gives an overview of when and when not to use a Deep learning model. Dataset collection is a big challenge and deep learning models rely heavily on it; a work around to this challenge is employed by implementing a naïve technique. This model requires almost no data compared to that of a Deep learning model and performs better. The result of this report will give a solution to automate the task of manually writing down the characters of a license plate. The proposed solution can be used at campuses, universities, airports, shopping complexes and hotels for intelligent surveillance.

Keywords: Computer Vision, Deep learning, Image Recognition.

1 Introduction

With recent advancement in intelligent transportation systems, Automatic license plate detection and recognition has become an interesting field of research. This work introduces the concept of computer aided license plate character recognition. This work gives a solution to automate the task of manually writing down the characters of a license plate. We have divided the problem into 3 parts – detection of the license plate, segmentation of the characters in the license plate and lastly recognition of these characters.

There are many ways these problems can be solved. Traditionally detection of objects was done through an edge detection algorithm. The problem with these algorithms was that they are not able

to generalize. Not because they were flawed but because there is a lot of variation in the images, which made it impossible for one algorithm to generalize on all the images. However, there are certain algorithms that work exceptionally well but due to high computational complexity these algorithms are not able to produce real time results.

We must make a system that is both accurate and fast at the same time. Deep learning technology helps us achieve that. Not only is it fast but also extremely accurate, if given enough data.

However, our entire pipeline will not solely be built on deep learning as we don't have enough good quality data. Brief overview of the pipeline: -

YOLO v3 is being used for the detection of license plates. This network is trained on 700 images of cars that contain Tunisian license plates. This same model when tested on Indian license plate gave around 97% accuracy. After the plate is detected it must be segmented and individual characters are extracted, this task is particularly difficult as there are no defined intervals or spacing between the characters of the license plate. So, we must use more dynamic methods or algorithms. Once the characters are extracted the next step is to recognize them, for that we have proposed two methods one is using the state-of-the-art CNN technique and the other is using matrix subtraction.

2 Problem Statement

This report basically targets three main challenges- detection of the license plate, Extraction of its characters and lastly recognition of these characters as shown in Figure 1.

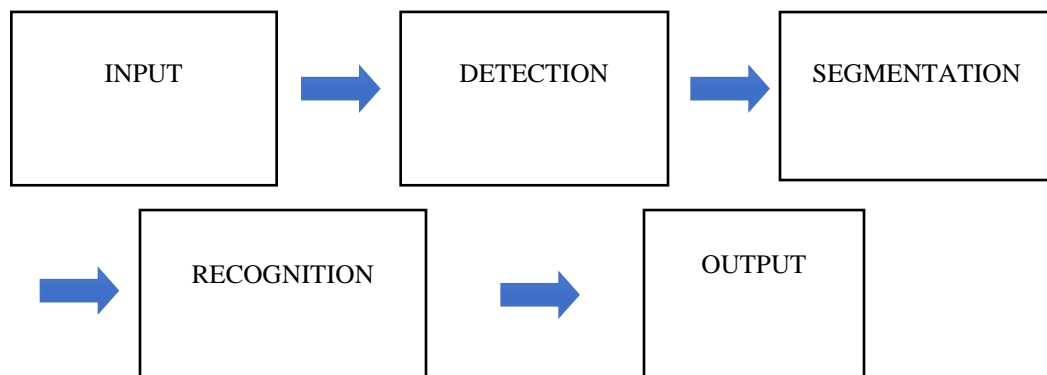


Fig. 1. Flow diagram representing the system pipeline

Detection here refers to the detection of license plates in an image, it is important as leads to compression of the data. Moreover, it helps in getting better accuracy when recognizing the characters because in this way we are sending only that information that we need to get recognized. You can imagine in an image there can be lots of different things apart from the license plate like sign boards, banners etc. So, to be specific we only send that data from the image that we want to get predicted.

Segmentation is done to extract out the individual characters from the detected license plate. Segmentation is not necessary as it depends on the implementation of the recognition model. As we are recognizing the characters individually therefore, we need to segment out characters.

Recognition of the final step in the pipeline, in this paper we have thrown light on two ways of recognition. First is using the state-of-the-art technique that is Deep learning and second is a naïve technique of matrix subtraction. We have discussed the merits and demerits of both the techniques.

3 Literature Review

Detection of the license plate is the first challenge in the process. Canny edge detection is one of the approaches that is used in a lot of tasks to detect edges in a scene. It was developed by John F. Canny in 1986. It is a technique to extract useful structural information from different vision objects and dramatically reduces the amount of data to be processed. Unlike traditional edge detection algorithms, the Canny edge detector performs better due to the presence of Gaussian filter, it removes any noise in an image. It is also immune to noisy environments and gives good localization.

This detection technique is likely to be used in the following way for license plate detection:

“1. Four Connected Points are searched from the image.

2. Width/Height ratio is matched against those connected points.

3. License Plate region is extracted from image.

4. Transformation of extracted license plate is performed.”

The success of this detection algorithm is strongly affected by environmental changes. For example, this detection algorithm might be able to detect certain objects in an image but might not give similar results on the same image taken from a different angle or in a different lighting condition. There is one more short coming of Canny edge detection algorithm: it is computationally expensive and can't deliver real time results.

Another approach for the detection of license plate is using Haar cascades(Haar Cascade is a machine learning object detection algorithm used to identify objects in an image or video and based on the concept of features proposed by Paul Viola and Michael Jones in their paper "Rapid Object

Detection using a Boosted Cascade of Simple Features" in 2001). A Haar-Like feature is just like a kernel in CNN, except that in a CNN, the values of the kernel are determined by training, while a Haar-Feature is manually determined. Adding more Haar-Features significantly requires lesser data for training and makes training faster. But there is a limit up to which one can hand-code Haar-features, as it is a tedious job. A slight change in the environment like change in the distance of an object from the camera or tilting of the object can make the Haar-classifier to fail.

The second step in the process is the recognition or the identification of the characters. OCR or Optical Character Recognition is the conversion of text in images into machine encoded format to make it understandable by a machine. It is widely used as a form of information 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. Traditional methods of OCR work very well for structured texts. The above-mentioned items have text of a very specific font style and size, that is why OCR works very well on these items. But there is a catch even with these items if the image is not taken appropriately, for example if the image is skewed or tilted this method of recognition is likely to fail.

In a real-life scenario, it is unlikely that the image taken will be ideal for Traditional OCR methods, especially in our context.

4 Work Done

The following sections will give an overview of the various methods that can be used in developing this pipeline. It will give you an idea of which method or approach works best in what case.

4.1.0 Detection

In this paper, we have used YOLO (You Only Look Once) a state-of-the-art object detection technique. It can perform real time detection with high precision. It is faster than traditional methods used for detecting objects in a scene.

Yolo is a single network trained end to end to perform a regression task predicting both objects bounding box and object class. It works on DarkNet framework (Darknet is a framework to train neural networks).

Since its features are learned from training on images, it generalizes very well on images it has not seen before. Unlike in Canny edge detection where the algorithm focuses on local changes YOLO looks at the image at once which gives it full understanding of the semantics of the image.

4.1.1 Working of YOLO

Unlike other methods Yolo looks at the image “only once”. It then divides the image into $S \times S$ grid, within each grid it takes “ m ” bounding boxes, the network outputs class probability for each of the bounding box and an offset value for the bounding box. The bounding boxes having class probability over a certain threshold value are selected and used to locate the object within the image. For example, in the image shown (Figure 2. [7]) below the network has predicted the various objects in the scene.

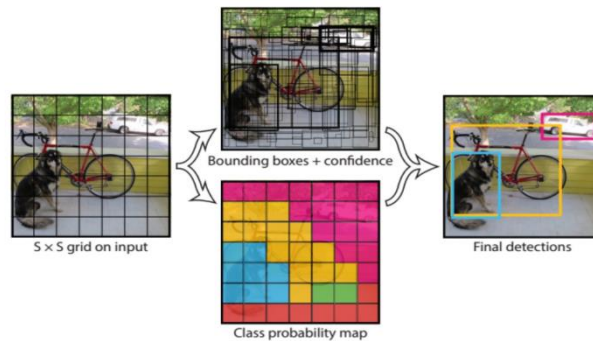


Fig. 2. Figuring showing how yolo technique works

It works on both images taken in daylight and night Figure 3. a,b,c . Also, the distance of the license plate from the camera doesn't matter too much.

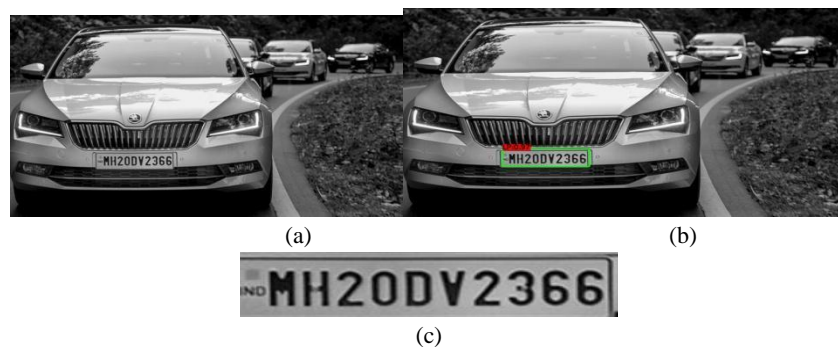


Fig. 3. (a) Input image (b) detected plate (c) extracted license plate in daylight



Fig. 4. (a) Input image (b) picture of extracted license plate at night

4.1.2 Advantages of YOLO

- Faster than traditional approaches
- Can Achieve Real time (up to 40 fps on GPU)
- Accurately detects license plates in different environmental conditions.
- Can detect multiple license plates in an image (*).

4.2.0 License plate segmentation

There are two ways to use the extracted license plate for recognition. First one is to directly put the entire extracted image into a neural network for prediction. This approach can be used if we have a huge dataset, which we don't have. Second way is to extract out individual characters from the extracted image and use those individuals' characters or segmented characters for recognition. We can use a pre-trained network that is trained on MNIST dataset or we can use a naive method, we will discuss the accuracy of both the approaches later in the paper.

Steps Taken before segmentation

- 1) Take Compliment
- 2) BGR to Gray
- 3) Binarization
- 4) Blur

4.2.1 The histogram of pixel projection

It consists of finding the upper and lower limits, left and right of each character. We perform a horizontal projection to find the top and bottom positions of the characters. The value of a group of histograms is the sum of the white pixels along a line in the horizontal direction. When all the values along all the lines in the horizontal direction are calculated, the horizontal projection histogram is obtained. The average value of the histogram is then used as a threshold to determine the upper and lower limits. The central area whose segment of the histogram is greater than the threshold is recorded as the area delimited by the upper and lower limits. Then in the same way we calculate the

vertical projection histogram but by changing the rows by the columns of the image to have the two limits of each character (left and right).[8]



Fig. 5. Segmented license plate

These green rectangular boxes represent the segmented characters as shown in Figure 5. and can be stored in an array for recognition.

4.3.0 License plate Character Recognition

This is the core of our system, as this will give us the final predictions. We have discussed two strategies for this task.

4.3.1 Recognition with CNN (Deep Learning)

The task to recognize the characters can be divided into two parts: -

- Digits Recognition
- Alphabet Recognition

Both above tasks can have separate machine learning models or one. In this paper, we have used two different deep learning models for the license plate recognition. One model is used to recognize the digits and the other recognizes alphabets. This will help in improving the accuracy.

Why? Because firstly we don't have enough data to train a model, if we use a single model then it will have to predict 36 classes. For each class we must have 500 to 1000 images. Moreover, certain letters like "S" and "5", "O" and "0" look almost the same. To make the model able to classify these classes, we must give enough images of 5's and S's, 0's and O's so that the model is able to distinguish between these classes.

On the other hand, if we use two different models then we will not have the above problem, which will let the model train on less data and still produce good results.

The high security number plates follow a certain standard thus we can assume at what places an alphabet or a digit can occur, and so we can use the various models for prediction accordingly.

Performance of the Deep learning model:

Table 1. The models used for predictions are basically handwritten character recognition models.

Detected/segmented	Actual	predicted
 	MH 20 DV 2366	MH 20 VU 2352
 	MH 14 GA 5284	MH 70 FO 5230
 	MH 20 EJ 0364	MH 20 EJ 04_4 <i>(connected 3 & 6 due the screw head that joined the two digits)</i>

The result is quite good given that we have used those pre-trained networks that are not specifically trained on license plate characters. However, given our objective this has a very low accuracy. We retrained the model based on transfer learning but got the same results. As we don't have enough data, for example there are classes which have more than 100 images and others have less than 10 and some of the classes are missing from the dataset. The count of our classes that we are predicting is 36 (Alphabets -> 26, Digits -> 10) which is quite a big number given the amount of the dataset we have for training. Below are two figs of model accuracy and Model loss of the model that we trained.

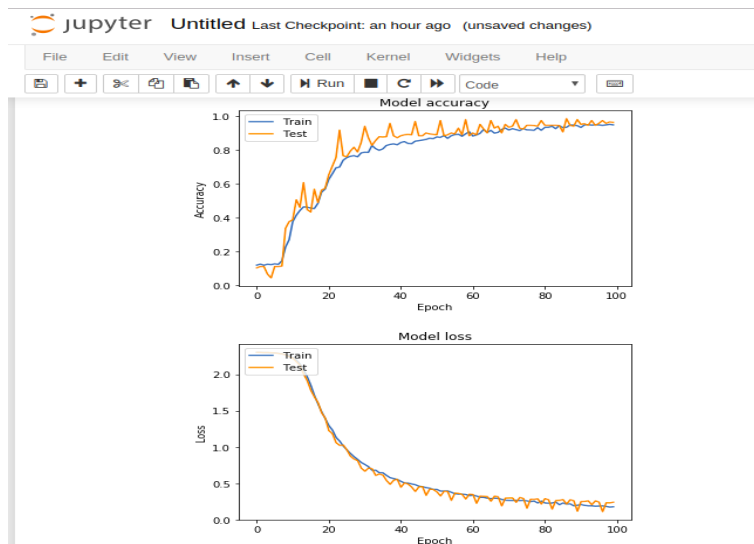


Fig. 6. Model accuracy and Model loss

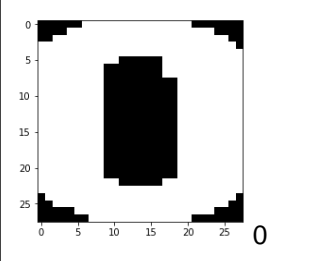
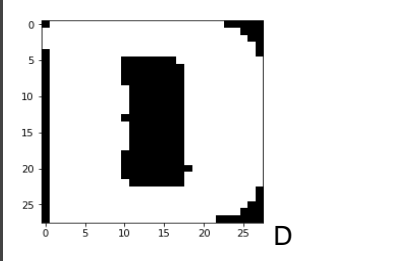
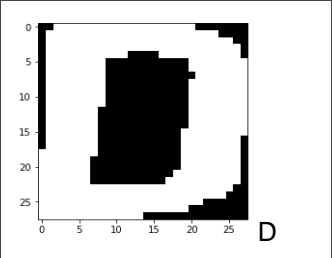
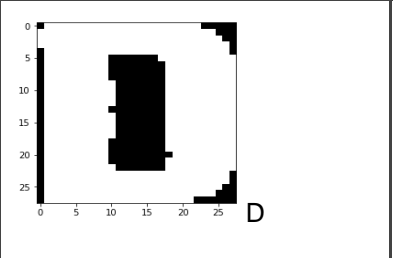
There is a lot of fluctuation in the graph, which means the model is overfitting. It is due to the reason we do not have a big enough dataset for training. So, we need more data, we can click more images of cars in a parking lot and then manually segment each character out of the images and use it to train our network.

The other is we can consider that all the number plates will be high security number plates, that implies that the font and size of the characters will be the same throughout. Now we will have the same characters with a lot less variation between them. So we can use a method that requires less or no data at all.

4.3.2 Matrix subtraction

Since to train a deep learning model we need a lot of data, at least 1000 images of each of the 37 classes. We can go with a naive approach that will save us from a lot of tedious work to gather thousands of images. All we need is just one image of each class in the dataset (call it as templates). Then what we will do is take our segmented images one by one and will subtract it from each of the template images and the image that will have the least difference will be our prediction.

Table 2 The images on the left are the segmented image and on the right are templates. The result will be D







	-		= 8.538
	-		= 3.714

This approach is quite useful in this scenario particularly as there is not a lot of variation in the font of standard number plate.

Performance of template matching:

With this approach along with getting better results we did also get rid of manually specifying the places of alphabets and digits.

Table 3 The models used for predictions are basically handwritten character recognition models.

Detected/segmented	Actual	predicted
 	MH 20 DV 2366	MH 20DV 2366
 	MH 14 GA 5284	MH 14 GA 528A
 	MH 20 EJ 0364	BH D0 E0 0MM_ (connected 3 & 6 due the screw head that joined the two digits)

5 Conclusion

Detection part of license plate in the scene is very robust. Factors like Distance, lighting conditions, skewness, and position of the number plate in the scene have almost no effect on the detection

system. The recognition part can be improved by adding more data points to the dataset. We can also go with the transfer learning approach, which means that we will use a already trained CNN model and will retrain some of its layers (exact number of layers depends upon various factors like amount of data we have and how close is your particular problem related to the pre-trained model) on the dataset we have with us. This approach will allow the network to generalize, so the model doesn't over fit.

The matrix subtraction approach overcomes the problem of small dataset by using a naive method. It was a simple approach with the least computational time. This approach proved to be useful in our case as there is not a lot of variation in the images that we are trying to predict. This method can be used as a temporary replacement until we got enough data for our deep learning model to train on.

6 Future Scope

For the detection part, the system is robust there are certain cases where it might not be able to detect the license plate, but those cases are almost negligible. The recognition part is where major improvements can be made. Rather than using a model trained on Handwritten MNIST dataset it should be trained on actual number plates letters. Since the letters in a license plate in a specific country have the same font throughout the country. It will not require a huge dataset since the variations will be a lot less but there should be enough samples for each class. It is obvious that not all license plates follow the same standard due to lack of strict laws, so hard coding the positions for the alphabets and digits seems obsolete. A solution to that can be to use just one model but then it would need a lot of data and cases like 'S' and '5' will still be a big problem in this case. Matrix subtraction performed better than the deep learning on the images we tested. It is supposed to perform better if the license plate follows a certain standard.

For now, we will be using the matrix subtraction method until we get data for training our deep learning model but because matrix subtraction method is not learning anything from the data it is given, it is just basically subtracting one matrix from the other, so in this approach there is no room for improvement. If it is predicting something wrongly there is no way, we can modify it so that it changes its prediction and give the right result. Although, certain modifications in the template or the input image can be done but still it might not generalize on other images. On the other hand, A Deep learning model tries learning features from the data it is given and if some results are wrong it has a "backpropagation" mechanism, which updates the model and after enough training it can learn from its mistakes and give right results. In this method, there is no upper limit to how much data we can provide to the model and train it. There will always be removal for improvement.

7 References

- [1] Uma, S., and M. Sharmila. "Implementation of License Plate Recognition System in ARM Cortex A8 Board." *IJCEM Int. J. Comput. Eng. Manag* 19.3 (2016): 9-13.
- [2] Zheng, Qiangqing, and Xiaolan Xie. "Traffic Sign Recognition Based on Learning Vector Quantization and Convolution Neural Network." *Proceedings of the 3rd International Conference on Intelligent Information Processing*. ACM, 2018.
- [3] Lin, Moule, Changxi Chen, and Chunchi Lai. "Object detection algorithm based AdaBoost residual correction Fast R-CNN on network." *Proceedings of the 2019 3rd International Conference on Deep Learning Technologies*. ACM, 2019.
- [4] Antipov, Grigory, et al. "Learned vs. hand-crafted features for pedestrian gender recognition." *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015.
- [5] Chen, Rung-Ching. "Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning." *Image and Vision Computing* 87 (2019): 47-56.
- [6] Laroca, Rayson, et al. "A robust real-time automatic license plate recognition based on the YOLO detector." *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018.
- [7] Anonymous, <https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006> - image (1-october-2019)
- [8] Anonymous, <https://towardsdatascience.com/automatic-license-plate-detection-recognition-using-deep-learning-624def07eaaf> - image (15-october-2019)
- [9] Anonymous, Source of the weights for the Detection model- <https://www.kaggle.com/achrafkhazri/yolo-weights-for-licence-plate-detector> (15-october-2019)