Term Project Report (CS555/CS455)

TOPIC : LICENSE PLATE DETECTION

GROUP MEMBERS :

- Rana Ertem (CS455)
- Swapnil Kharche (CS555)

(I) PROBLEM STATEMENT (ABSTRACT):

Due to a significant rise in the number of vehicles these days, the number of incidents involving stealing of vehicles, traffic rule violation and even accidents have increased. The License Plate Detection plays a significant role in all these scenarios and can also be used for numerous applications such as automatic toll collection, traffic law enforcement, parking lot access control and road traffic monitoring. Our project aims at reading the vehicle registration plate so that the data extracted from the plate can be stored and used for the above mentioned applications. In our project we use connected component analysis, segmentation and machine learning for character detection and recognition.

(II) LITERATURE REVIEW (INTRODUCTION):

Computer Vision and character recognition, algorithms for license plate recognition play an important role in video analysis of the number of the number plate image. A number of researches have been carried out to identify the type of vehicles and detect their plates. License Plates have different sizes, character formats and the color standards all over the world. However, generally license plates are characterized by high contrast between characters and underlying background. Many existing license plate recognition systems use Optical Character

Recognition (OCR) to read the data from the license plate. The early systems suffered from low recognition rate due to limited level of recognition software and vision hardware. Recent improvements in technology like infrared imaging and high resolution cameras have improved the recognition process. As mentioned earlier there are a number of other challenges that while detecting a license plate for example, detecting the region of interest, different standards, Noise that is introduced from the camera capture, etc. The complexity of each of these challenges determine the accuracy of the system. There are several new techniques that help to automatically detect license plates of a vehicle depending on the type of vehicle. For example using sobel filters and Support Vector Machines to determine the edges of the vehicle/plate and detect the region of interest, and then using OCR to determine the characters of the plate. However this techniques cannot be applied to a video source. Each of these techniques has its own pros and cons.

(III) METHOD (ALGORITHM):

Our project is based on the common pattern matching technique for the recognition of a single font and a fixed size character. We first determine the region of interest i.e the license plate using assumptions on the characteristics of the license plate. And then we use machine learning classifiers to compare common patterns for fixed size letters obtained using segmentation performed on the license plate. However, to achieve good performance, large amounts of samples are needed. We have used the scikit-image, an image processing package for python and scikit-learn, a machine learning package for python. The algorithm we have followed is as shown below:

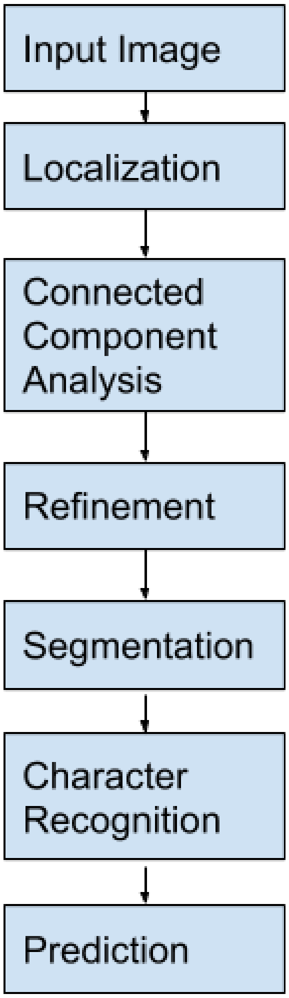


Figure 1: Algorithm

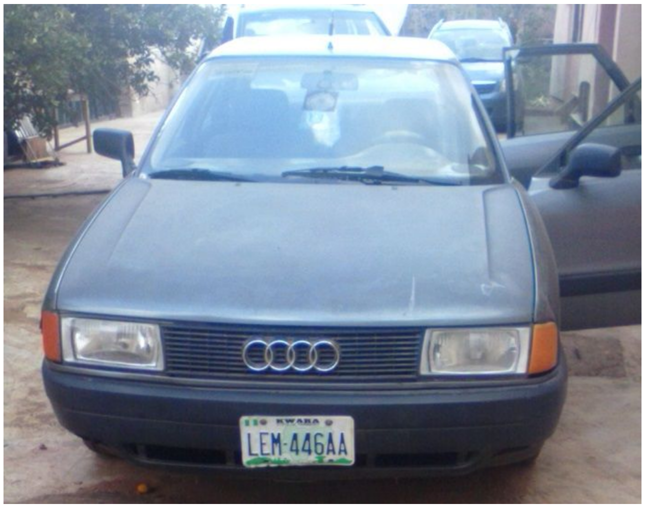


Figure 2: Input Image

1. LOCALIZATION (PREPROCESSING)

Generally it is too complex to detect target from a color image. In order to identify the outline of the vehicle/license plate easily, we first convert the input image to grayscale with each pixel being between 0 and 255. In many cases the luminance of the license plate can be brighter than other ordinary objects around vehicle in the image. Also sometimes, the contrast between license plate and other objects may not be sharp, which will decrease the accuracy of recognition if we try to extract the license plate directly. In such cases we can also stretch the intensity value of the grayscale image in order to increase the contrast of the output image. And since we are going to use the connected component analysis in the next step we then use the adaptive thresholding technique to convert it to a binary image. The output we get is as follows:

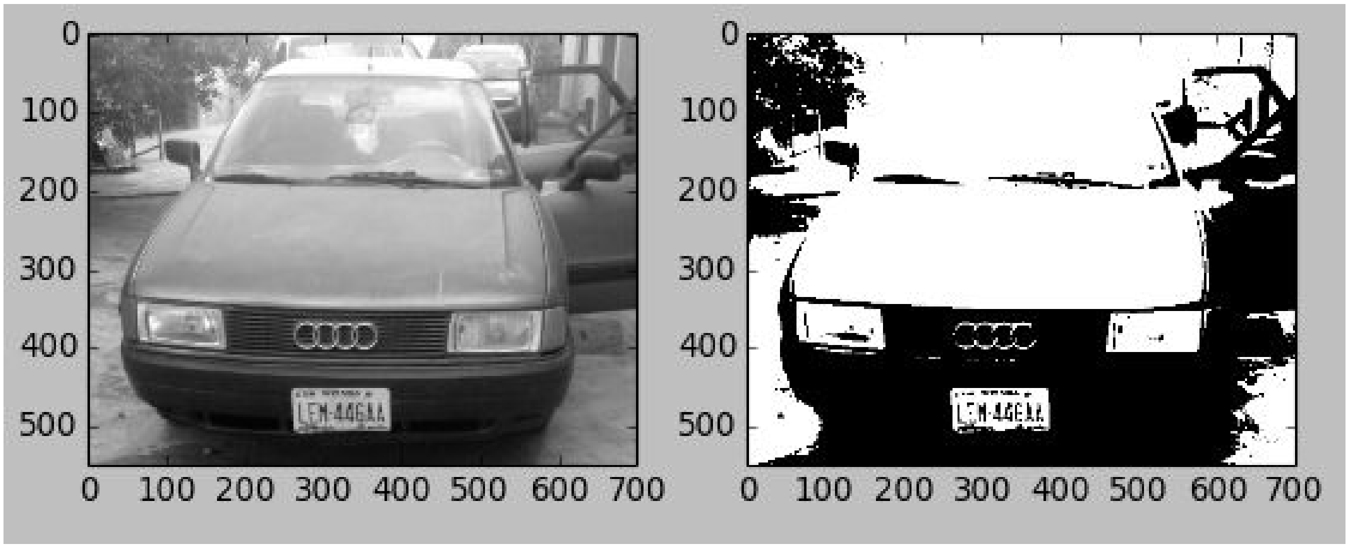


Figure 3: Preprocessing

2. CONNECTED COMPONENT ANALYSIS

In this step we use the concept of connected component analysis, which basically helps us group and label connected regions on the foreground. A pixel is deemed to be connected to another if they both have the same value and are adjacent to each other. As we know a license plate is usually in high contrast to the letters on it, so using the connected component analysis, mapping all the regions and labelling them would give us a possible position of the license plate within the image. Other methods like edge detection along with density of pixels and morphological operations can also help in determining the position of the license plate. The output we get after performing connected component analysis is as follows:

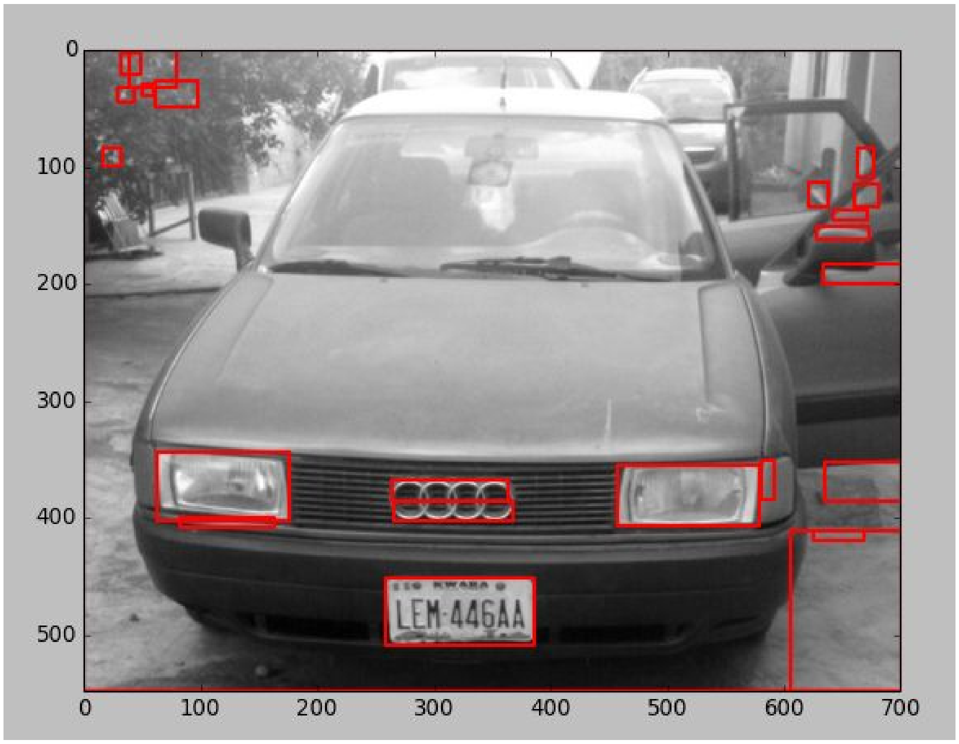


Figure 4: Connected Component Analysis

3. IMAGE REFINEMENT

As seen from the above image the connected component analysis gives us a number of regions that do not even contain the license plate. In order to eliminate such regions we must take into account the typical characteristics of a license plate. However, this can be a challenge, since as mentioned earlier, the license plate characteristics can vary a lot depending on the country, vehicle type, shape of the license plate and a number of other factors. For the purpose of the project we have made a few assumptions of our own, and these assumptions can be tweaked for a different environment. The assumptions are as follows:

a. The license plates are rectangular in shape, i.e the width is greater than the height.

b. The proportion of width of the license plate region with respect to the whole image is between 15% to 40%.

c. The proportion of height of the license plate region with respect to the whole image is between 8% to 20%.

Using these assumptions we are able to eliminate the regions which are not the license plate and we get the following output:

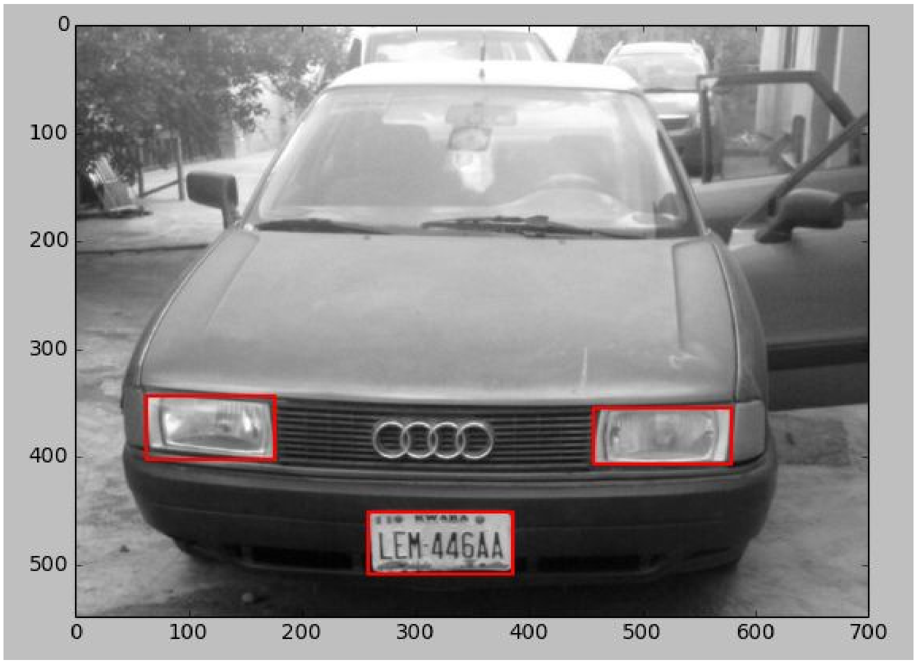


Figure 5: Image Refinement

However it is still possible that regions like headlamps/stickers/etc. may still be counted as a license plate. Such regions can be eliminated using the concept of vertical projection. It is implemented by adding all the pixels in each column and since the license plate area has lots of pixel values due to the characters on it, that region would be kept and all the other regions eliminated. We however have checked all the remaining regions with bounded boxes and the one which has the license plate actually prints out the image.

4. CHARACTER SEGMENTATION

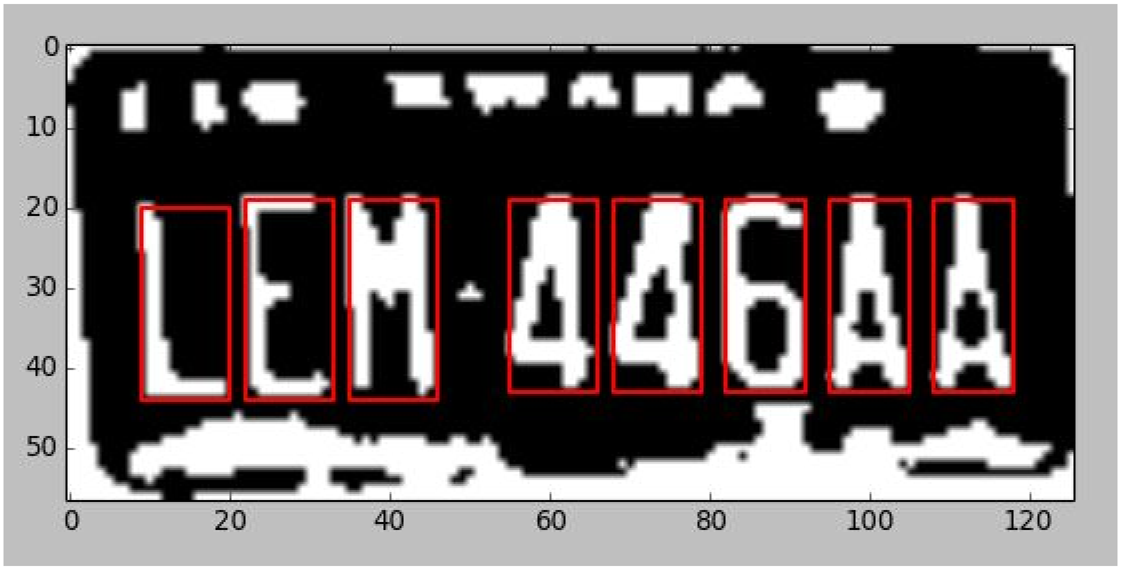


Figure 6: Character Segmentation

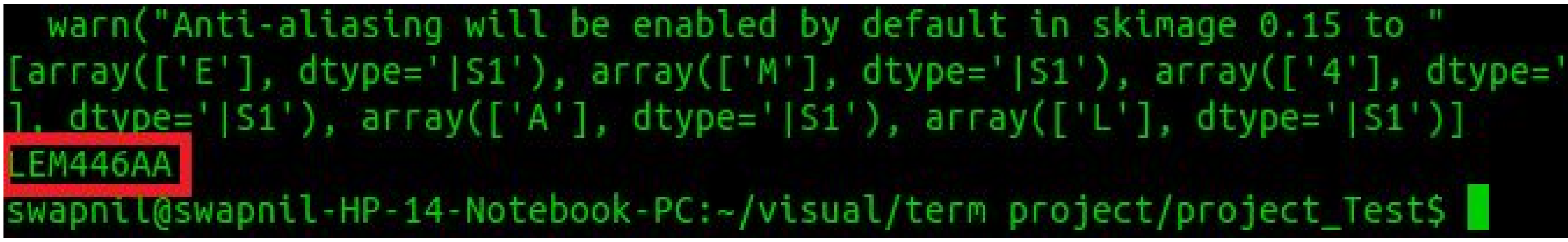
Once we have the license plate we map out all the characters of the license plate. Again using connected component analysis each character was segmented and resized to a 20px by 20px. In order to get a more clearer image of the characters morphological operations like erosion and dilation can be performed. A data structure is kept to maintain the order of the characters.

5. CHARACTER RECOGNITION

For character recognition we use the Classification Supervised Learning. We have chosen a training dataset and a supervised learning classifier (Support Vector Classifiers). We then trained the chosen model, tested it out to see how accurate it was, and then finally used it for the actual prediction. Our training dataset consists of a a set of ten 20px by 20px images of each of the alphabets and numbers. As mentioned earlier we have used scikit-learn, a python based machine learning package to train the model. Once we have trained the model we perform a 4-cross validation on model to check it’s accuracy, i.e we used 3/4th of the dataset for training and 1/4th of it for testing. And finally the model was persisted to a file so that predictions can be made without the need for training the model again.

(IV) RESULTS/CHARACTER PREDICTION

Once we have the trained model we attempt to predict the characters on the license plate and we get the following output (highlighted in red):



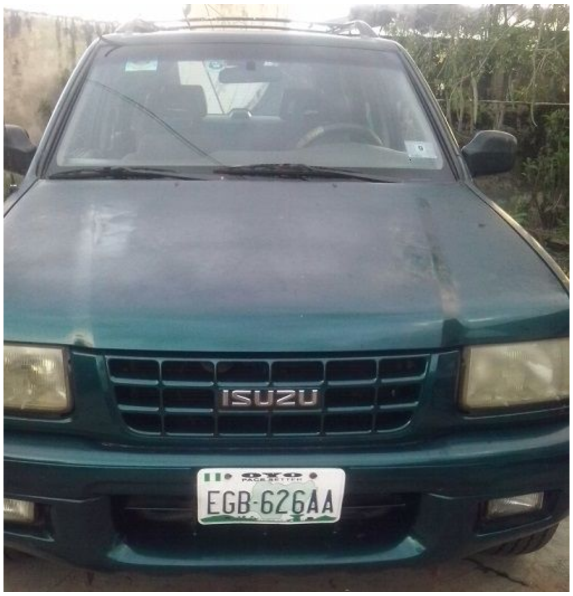
```
  warn("Anti-aliasing will be enabled by default in skimage 0.15 to "
[array(['E'], dtype='|S1'), array(['M'], dtype='|S1'), array(['4'], dtype='
], dtype='|S1'), array(['A'], dtype='|S1'), array(['L'], dtype='|S1')]
LEM446AA
swapnil@swapnil-HP-14-Notebook-PC:~/visual/term project/project_Test$
```

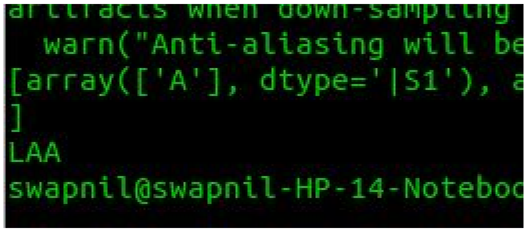
As we can see the predicted characters are the same as the characters of the license plate from the input image. The characters recognition happens out of order but since we have maintained a data structure to help us maintain the order of the characters, we get the correct sequence for the output.

We tested out the system on a number of images having different parameters, the following table summarizes the results for the same:

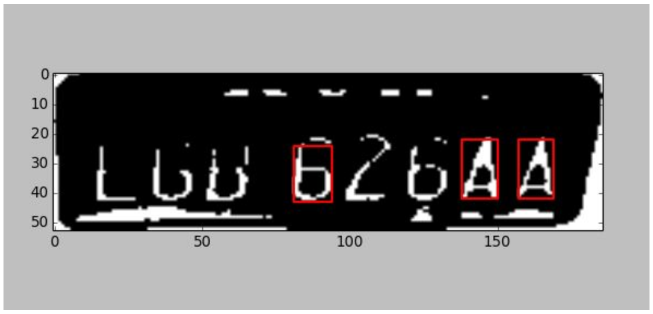
	Input Images	Success	Error	Accuracy
Distance between view and vehicle adjustment	10	7	3	70%
Angle of License Plate Adjustment	10	8	2	80%
Character Segmentation	10	7	3	70%
Character Recognition	10	10	0	100%

Some cases in which the implemented system fails are shown below:

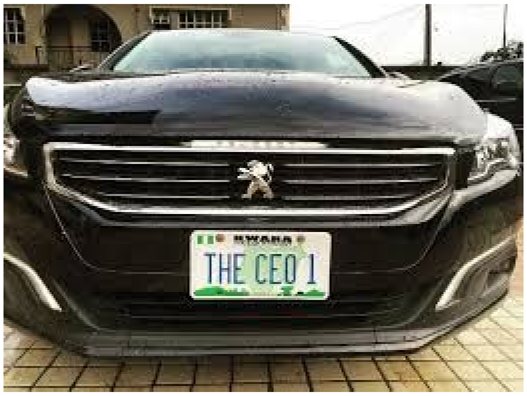


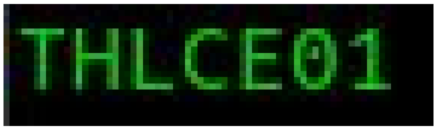


```
artifacts when down-sampling
  warn("Anti-aliasing will be
[array(['A'], dtype='|S1'), a
]
LAA
swapnil@swapnil-HP-14-Noteboo
```



As seen in the above cases we see that only the characters ‘6’, ‘A’ and ‘A’ are identified.





```
THLCE01
```

And as seen from the above case, the letter ‘E’ is identified as ‘L’.

(V) CONCLUSION/DISCUSSION:

LIMITATIONS:

1. As we can see above, when the Distance between vehicle and the viewer is adjusted the accuracy is altered, we noticed that for the system to work correctly the size of the image shouldn’t be too small or big, a width of approximately 600px works.
2. Also when the angle of the license plate is adjusted, depending on the brightness or contrast with the background, in some cases the system fails to get a perfectly thresholded image for the license plate and hence the connected component analysis doesn’t recognize the plate.
3. Character Segmentation works fine if the thresholded image has a clearly visible license plate, but in the cases mentioned above the segmentation may fail.

CONCLUSION:

Hence we can say that the system works fine in most of the cases, but in some cases as mentioned above there maybe some error. However, the main aim of the system is accomplished and it may help in a number of real life applications as mentioned earlier.

FUTURE SCOPE:

Since the accuracy of the recognition is important so the system should be optimized and modified for overcoming the accuracy limitations. In order to make the recognition more precise we should add some preprocesses to remove interferences. Moreover we should test out the system in more complicated environments such as dark nights or heavy rains. If we could accomplish all these objectives the system would be fully complete.

(V) REFERENCES

1. *License Plate Recognition* by Songke Li & Yixian Chen, Faculty of Engineering and Sustainable Development, University of Gävle.
2. *Vehicle Number Plate Recognition System* by Aniruddh Puranic, Deepak K. T and Umadevi V., Dept. of CSE, BMSCE.
3. *Automatic License and Number Plate Recognition System for Vehicle Identification* by Hamed Saghaei Department of Electrical Engineering, Faculty of Engineering Shahrekord Branch, Islamic Azad University
4. https://blog.devcenter.co/developing-a-license-plate-recognition-system-with-machine-learning-in-python-787833569ccd