Number Plate Text Recognition in Real Time

Under Supervision Of
Prof. Harish Karnick

Abhay Kumar (11008)
Amit Kumar (12096)
Aradhana Singh (12150)
Satyam Kumar Shivam (11917658)
Indian Institute of Technology, Kanpur

April 15, 2016

Abstract

The aim of this project is to detect the number plates of four wheeler vehicles coming and going through in given surveillance video data and read the numbers on the detected number plate. Our procedure starts with extracting frames from the given data. Image enhancement and noise removing techniques are implemented in order to improve the quality of images. We used K-nearest neighbour model to recognise the appropriate character in the image. Based on the characters recognized in the image, possible plates are detected. The plate with maximum number of characters is annotated on the car in the image. We repeated this procedure on several video files in order to evaluate our model performance.

Acknowledgement

We are grateful to everyone who supported us throughout the course of this project. We are thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. We express our sincere gratitude towards our Prof. Harish Karnick whose consistent efforts, diligent support and regular encouragement helped us to overcome the challenges that we faced throughout this project and to acquire detailed knowledge in the concerned topic.

Contents

1

Introduction

Number plate text recognition in real time in traffic video is a challenging task and has many real life applications such as smart traffic monitoring, e-toll collection, vehicle tracking, etc. With this project, we aim to approach this problem and obtain accurate results and annotate with the predicted plate's numbers on the vehicles in real time. We are targeting only for four wheeler vehicles. The annotation is done on the video data itself as the final output. Firstly, we extract frames from the given video and then pre-process it with histogram equalization and median filtering. After pre-process, we localize the possible region of interest and then apply k-nn based classifier to recognize characters present in the extracted license plate region. For this, we have used Python and Open Computer Vision (OpenCV) library in order to use existing computer vision tools. We have tried to make our algorithm to perform in real time by using parallel computations and multithreading.

2

Methodology

We have proposed a method which is used to recognise license plate character on four wheeler vehicles. Indian vehicles have most often licence plates with English characters but very few of them also use Devanagari character on licence plates. We have confined training of our model on English characters and digit data-set only. Below is the detailed description of the methodology we have used in the process.

2.1 Video Capture

Video of the moving vehicle is captured using a high quality camera. Capturing the frames, with plates that clearly show a licence number, is heavily affected by position of the camera.

2.2 Frame Extraction and Enhancement

Frames are extracted from the video at a regular interval of 10 millisecond. Frame are further enhanced to remove the noise visual data by applying median filter. Histogram equalization is performed on frames for contrast adjustment.

2.3 k-nn Classifier Training

k-nn classifier is used in OCR algorithm for the label prediction of character images. We have trained classifier using printed digit images with a labelled data. Images data has been saved as flattened array for training. Training is done for different values of k for improving accuracy in testing data-set.

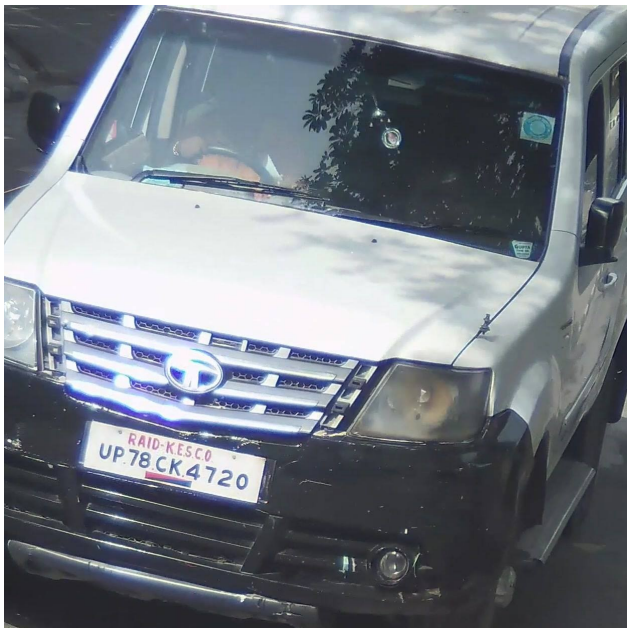

Figure 2.1: Original Image

Pixel density is used to compute similarity between two images while training k-nn classifier.

2.4 Character Detection in Image

Image thresholding is done for extracted frame. We locate all the contours in the image and then look for possible match with a character, if get some possibility then we add it to our list of possible characters. We are using criterion like minimum pixel area and pixel width for possibility prediction. Note at this point we vaguely know that this could be a character but still unaware of the exact label of it.

2.5 Character Recognition

This step predict the label for the found contours in the previous step. This predict the label on best match of character image from the learned representation of characters.

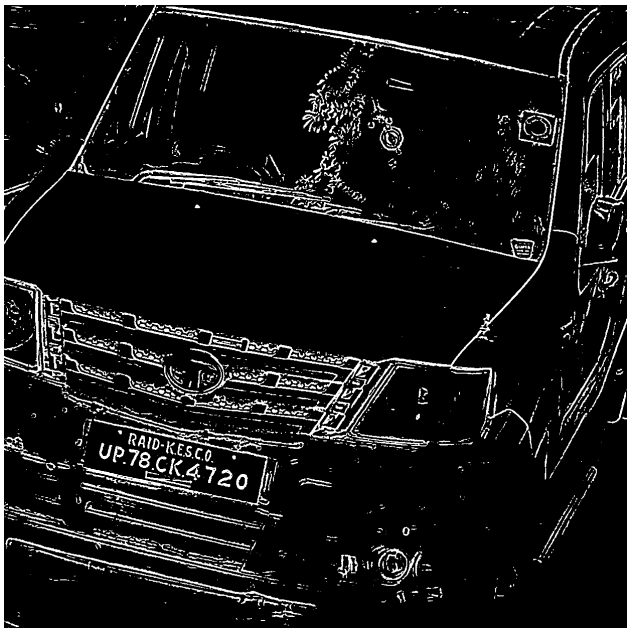

Figure 2.2: Threshold Image with contours

2.6 Possible Plates Detection

Possible plates are found based on the characters group found together. List is then sorted giving more priority to the one have highest number of characters.

2.7 Licence Number Plate prediction for Each Frame

Based on all the plates detected in a image frame. Character string with largest size is predicted as the final output for the possible licence plate found in that frame. This result may not be able to accurately get the full plates but using the series of frame we can predict the most accurate full string of characters.

2.8 Confidence Score for Each Frame

We have used similarity matrix based on Longest common sub-sequence among the last five frames. Using this matrix we are predicting the best string found among the last five frames. We have made a very powerful as-

sumption that a sequence of frame each containing quite similar features and predicting long character string with high confidence score, will aggregate to the correct string of characters.

2.9 Parallel Computation

We have also attempted implementing multi-threading in our algorithm for fast computation to make our algorithm work in real time. A little improvement in computation time is observed after implementing multi-threading.

3

Algorithm

1. Extract frames from video.
2. for each extracted frame:
   (a) Convert from RGB to grey scale.
   (b) Apply median filtering and histogram equalization.
   (c) Apply thresholding.
   (d) Localize best plate region out of possible plate regions.
   (e) Apply edge detection and segment the localized plate.
   (f) Character recognition by using a trained k-nn classifier.
3. Obtain a 5X5 matrix having its element as score based on longest common sub-sequence of last 5 recognized plate numbers.
4. Normalize scores of the obtained matrix.
5. Choose best recognized number plate based on sum of scores corresponding to each last 5 recognized number plates.

4

Data-set

4.1 Training data-set

We have trained our k-nn classifier using printed digit images with labelled data as shown below.

Figure 4.1: Image containing training image

Training image data-set contains 180 images of characters of alphabets and numbers. Images data has been saved as flattened array for training purpose.

4.2 Testing data-set

Testing data-set contains surveillance videos that are provided to us by CS771A course ftp site.

5

Results

We obtained results corresponding to each surveillance video data provided to us. Some results of intermediate frames of given video data are shown below:

Figure 5.1: Result for sample frame obtained from video data

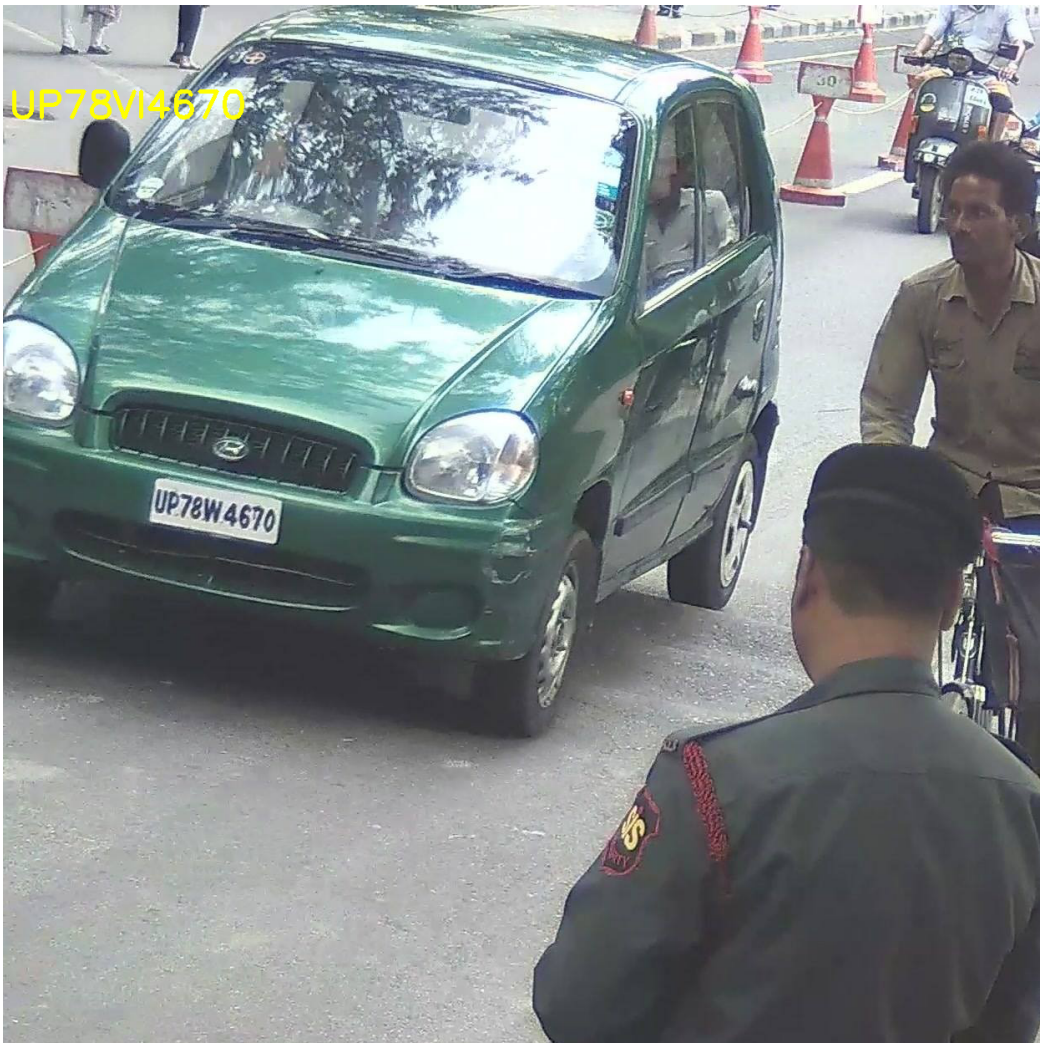

Figure 5.2: Result for sample frame obtained from video data

We computed accuracy for a frame based on number of characters correctly matched in that frame. We performed our algorithm on each video data 5 times so as to accurately evaluate our algorithm. Below is summary of the average accuracy obtained for each video data in the following table:

Video File Name	Chars Present	Chars Correctly Detected	Accuracy
input_video_sample2.mov	9	7.3	0.81
input_video_sample3.mov	10	9.2	0.92
videosample5.mov	120	73.4	0.61

Here, first column represents name of video file, second column represents number of actual characters (average of 5 iterations) present in license plate numbers of passing by four wheelers in the video data file and third column represents number of character correctly matched (average of 5 iterations) in best extracted license plate numbers of passing by four wheelers in that video data file.

On three video data files, average accuracy obtained is 0.78.

6

Conclusion

In this project, we have explored various approaches to recognize license plate numbers from video data in real time. After applying pre-processing steps on the frames of given video data, we approached character recognition problem by using a k-nearest neighbour based classifier. By adjusting different parameters, like aspect ratio, we are able to successfully recognize license plate numbers of four wheelers from the given video data. We found a trade-off between getting accuracy of recognized plate number vs making the algorithm run on real-time video data. For accuracy, we have implemented a longest common sub-sequence matrix based approach in which we compute longest common sub-sequence among plate number recognized in last 5 frames and then we assigned a confidence score to each plate number recognized. With this algorithm, we are able to recognize license plate number with fairly good accuracy and real-time performance.

Bibliography

[1] Wikipedia contributors. "Automatic number plate recognition." *Wikipedia, The Free Encyclopedia. Wikipedia*, The Free Encyclopedia, 15 Apr. 2016. Web. 15 Apr. 2016.

[2] Sajjad, K. M. "Automatic License Plate Recognition using Python and OpenCV." *Department of Computer Science and Engineering MES College of Engineering, Kuttippuram, Kerala, me@ sajjad. in* (2010).

[3] Roy, Sourav, Amitava Choudhury, and Joydeep Mukherjee. "An approach towards detection of indian number plate from vehicle." *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* 2.4 (2013): 241-244.

[4] Kulkarni, Prathamesh, et al. "Automatic Number Plate Recognition (ANPR) System for Indian conditions." Radioelektronika, 2009. *RADIOELEKTRONIKA'09. 19th International Conference.* IEEE, 2009.

[5] "MicrocontrollersAndMore/OpenCV3LicensePlateRecognitionPython." GitHub. N.p., n.d. Web. 15 Apr. 2016.