Number recognition using HD-MR algorithm

Submitted for the Course

ECE6045 Neural Networks & Fuzzy Systems
Winter Semester 2019-20

by

Hanita Jain – 19MDT0010
B.Samanvitha – 19MDT0043
G. Venkata Manish Reddy – 19MDT0052

Prof. S. Sankar Ganesh

School of Electronics and Engineering

VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

May, 2020

Table of Contents

Introduction

Electricity plays a major role in our lives. The use of electricity is increasing every day. We use it everywhere and for different purposes that we cannot imagine our lives without it. The tool used to measure the electricity consumption is Electric Meters.
Utility companies in developing countries employ analog electrical meters to determine consumption and bill their customers accordingly. Obtaining an accurate reading is an expensive and time-consuming process. High consumption levels of water, energy, or gas are fined by the government; thus, it is necessary to develop tools that allow users to be informed about their consumption in real time.

This work aims to facilitate the electricity meter reading mechanism for electricity company's employees, as the existing method of manual electric meter reading is not applicable with the increasing consumption of electricity and has a lot of disadvantages: It is very tedious, time consuming, man power consuming and is prone to lot of errors. We use a methodology based on image processing to obtain efficient and accurate reading of the digital electricity meters on the basis of license plate recognition algorithm. The study of this work is extracting a recognizing meter reading digits from electric meters.

Based on the collected data of electric meters, there are many versions of digital electric meter. In these versions the location of the reading area is on the top of the meter. In addition, they contain six digits of numbers in reading area. We take the help of a paper which proposes a new number recognition algorithm named the Hausdorff distance for meter reading (HD MR). Their experiments prove that HD MR can achieve a 99.9% recognition rate, even when recognized numbers are under rotation. Their maximum recognition time is 31 ms; hence, the proposed method proves to be effective and capable in real time for the task proposed.

The base of this HD MR algorithm is the vehicle plate recognition algorithm. Smart automated traffic enforcement solutions have been gaining popularity in recent years. These solutions are ubiquitously used for seat-belt violation detection, red-light violation detection and speed violation detection purposes. For example, in some cities in China, a new VPLR technology which enables drivers to pay parking fee using electronic wallet in a short time without leaving cars has received widely favorable reception. The parking fee from the drivers can be automatically collected by OCR (Optical Character Recognition) systems which can recognize license plates. However, errors sometimes occur when a vehicle is not identified or when a vehicle is wrongly identified as another vehicle. Highly accurate license plate recognition is an indispensable part of these systems. However, general license plate recognition systems require high resolution images for high performance.

However, the license plate can be easily affected by external factors such as lightning conditions, weather, and backgrounds; besides, most VPLR systems do not fully consider the complexity of background and illumination conditions in the practical application, so locating and detecting the license plate from original images accurately and efficiently are still vital steps and the main difficulties for successful license plate recognition.

Vehicle Plate Algorithm

Vehicle plate licenses recognition (VPLR) plays a significant role in the field of intelligent transportation system. It has been widely used in traffic management, vehicle monitoring, suspect vehicle tracking, and many other fields. Many new technologies like that are emerging and rising due to people's changing demands; it is necessary to improve license plate location and recognition algorithms to increase vehicle management efficiency. In general, license plate recognition process consists of three steps: license plate localization, characters segmentation, and characters classification and recognition. Since the license plate localization is the first and essential step of the recognition process, the result has a direct impact on the accuracy of character segmentation and character recognition. To recognize the vehicle number plates of the running vehicle is an exceedingly difficult task for a naked eye, for that License plate Recognition algorithm is used to capture the digits and letters from the image captured. This algorithm consists of 2 steps:

Step 1: The image is preprocessed and the location of the car, number plate and size of number plate are identified.

Step 2: A cropped image will be generated from the input image; it is converted into black and white style and the individual digits and letters are identified and displayed. This algorithm can also recognize the number plate of fast-moving car and number plates of side view with low accuracy

We have used manageable code in Python to detect the license plate and crop the recognized number plate image and store the image in a location using OpenCV and pytesseract (OCR) in order to extract the text (number plate) from the selected image.

Code

```
import numpy as np
import cv2
import imutils
import pytesseract
pytesseract.pytesseract.tesseract_cmd = r"C:\Program Files\Tesseract-
OCR\tesseract.exe"
# Read the image file
image = cv2.imread('Car Images/7.jpg')
# Resize the image - change width to 500
image = imutils.resize(image, width=500)
# Display the original image
cv2.imshow("Original Image", image)
cv2.waitKey(0)
# RGB to Gray scale conversion
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2.imshow("1 - Grayscale Conversion", gray)
cv2.waitKey(0)
# Noise removal with iterative bilateral filter(removes noise while preserving edges)
gray = cv2.bilateralFilter(gray, 11, 17, 17)
cv2.imshow("2 - Bilateral Filter", gray)
cv2.waitKey(0)

# Find Edges of the grayscale image
```

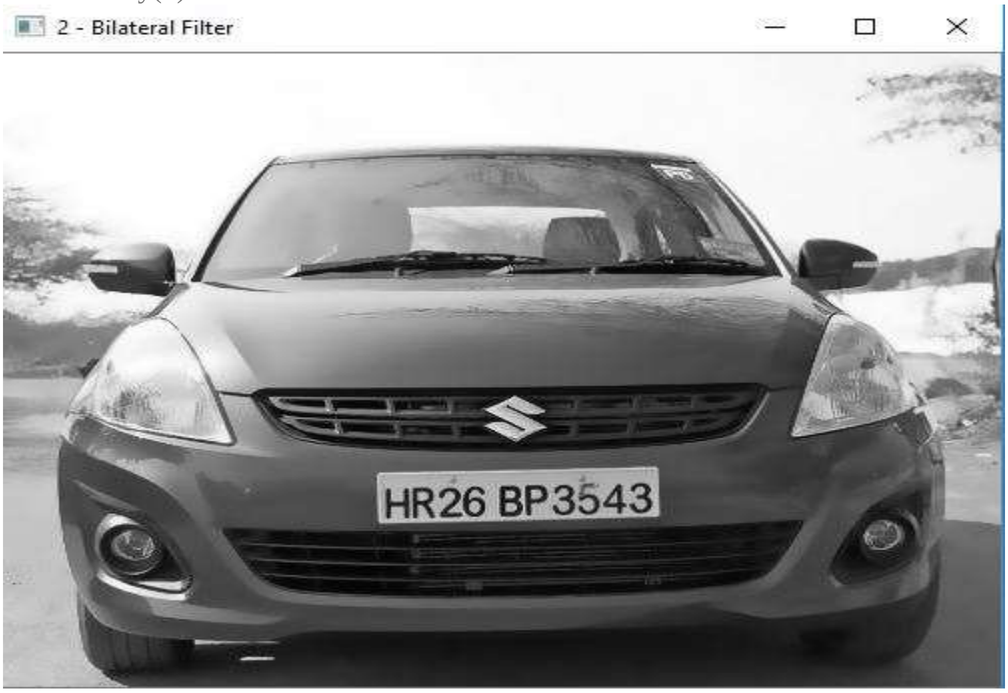
```
edged = cv2.Canny(gray, 170, 200)
cv2.imshow("3 - Canny Edges", edged)
cv2.waitKey(0)
# Find contours based on Edges
cnts, new  = cv2.findContours(edged.copy(), cv2.RETR_LIST,
cv2.CHAIN_APPROX_SIMPLE)
# Create copy of original image to draw all contours
img1 = image.copy()
cv2.drawContours(img1, cnts, -1, (0,255,0), 3)
cv2.imshow("4- All Contours", img1)
cv2.waitKey(0)
#sort contours based on their area keeping minimum required area as '30' (anything
smaller than this will not be considered)
cnts=sorted(cnts, key = cv2.contourArea, reverse = True)[:30]
NumberPlateCnt = None #we currently have no Number plate contour
# Top 30 Contours
img2 = image.copy()
cv2.drawContours(img2, cnts, -1, (0,255,0), 3)
cv2.imshow("5- Top 30 Contours", img2)
cv2.waitKey(0)
# loop over our contours to find the best possible approximate contour of number
plate
count = 0
idx =7
for c in cnts:
        peri = cv2.arcLength(c, True)
        approx = cv2.approxPolyDP(c, 0.02 * peri, True)
        # print ("approx = ",approx)
        if len(approx) == 4:  # Select the contour with 4 corners
            NumberPlateCnt = approx #This is our approx Number Plate Contour
            # Crop those contours and store it in Cropped Images folder
            x, y, w, h = cv2.boundingRect(c) #This will find out co-ord for plate
            new_img = gray[y:y + h, x:x + w] #Create new image
            cv2.imwrite('Cropped Images-Text/' + str(idx) + '.png', new_img)
#Store new image
            idx+=1
 break
# Drawing the selected contour on the original image
#print(NumberPlateCnt)
cv2.drawContours(image, [NumberPlateCnt], -1, (0,255,0), 3)
cv2.imshow("Final Image With Number Plate Detected", image)
cv2.waitKey(0)
Cropped_img_loc = 'Cropped Images-Text/7.png'
cv2.imshow("Cropped Image ", cv2.imread(Cropped_img_loc))

# Use tesseract to covert image into string
text = pytesseract.image_to_string(Cropped_img_loc, lang='eng')
print("Number is :", text)
cv2.waitKey(0) #Wait for user input before closing the images displayed
```

Results:

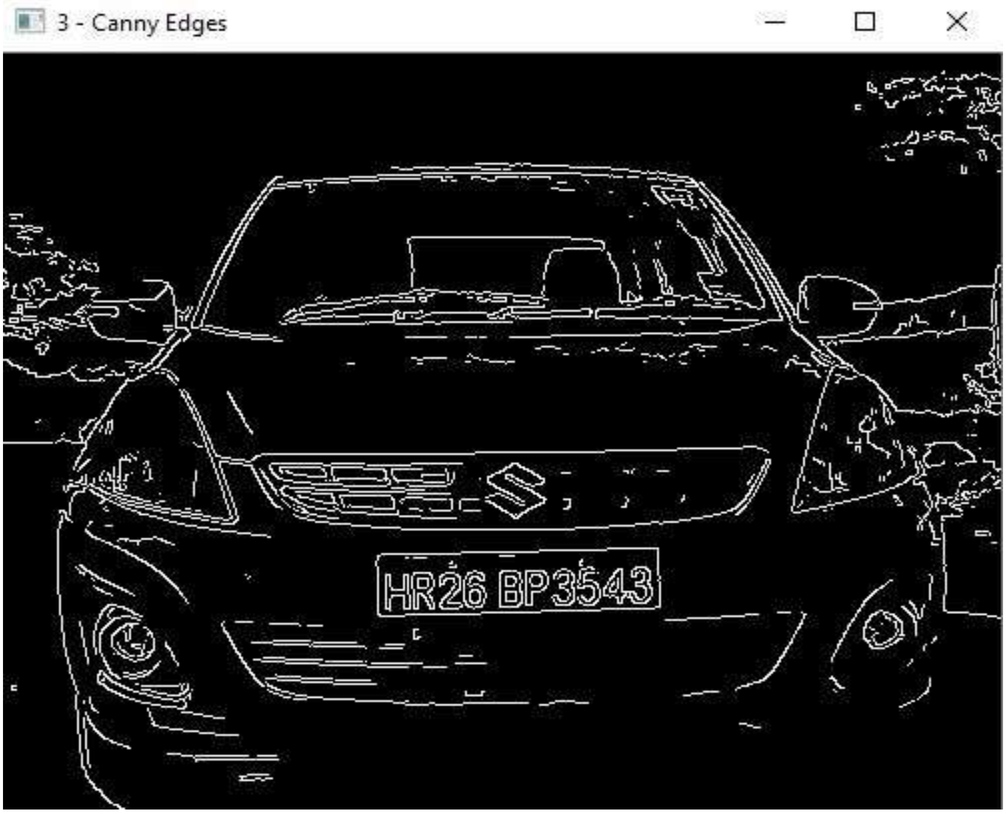
 # Display the original image
cv2.imshow("Original Image", image)
cv2.waitKey(0)


```
# RGB to Gray scale conversion
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2.imshow("1 - Grayscale Conversion", gray)
cv2.waitKey(0)
```

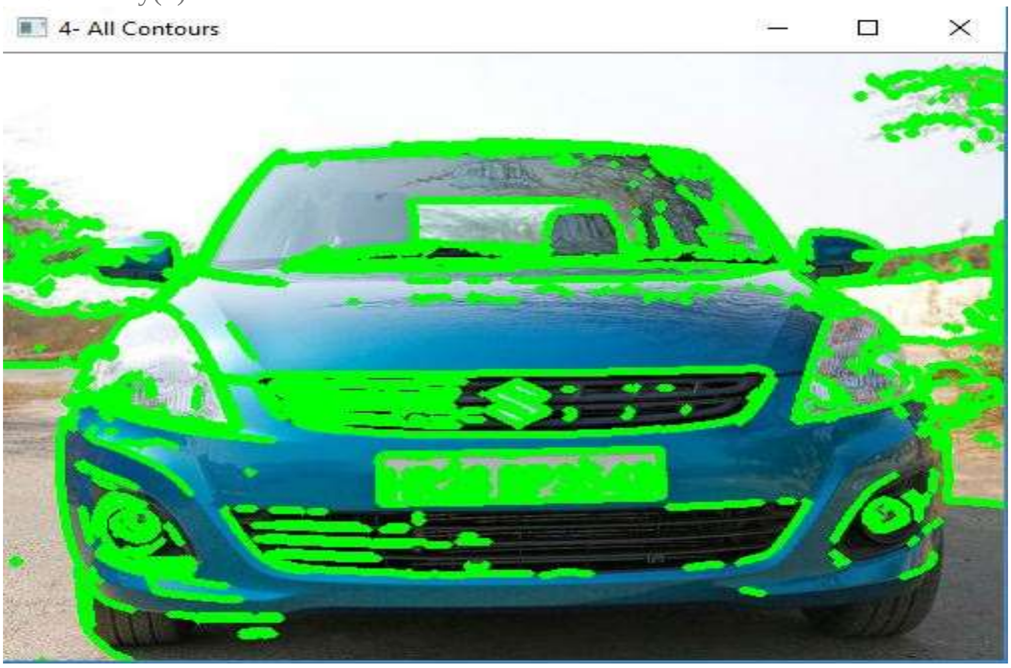
```
# Noise removal with iterative bilateral filter(removes noise while preserving edges)
gray = cv2.bilateralFilter(gray, 11, 17, 17)
cv2.imshow("2 - Bilateral Filter", gray)
cv2.waitKey(0)
```




```
# Find Edges of the grayscale image
edged = cv2.Canny(gray, 170, 200)
cv2.imshow("3 - Canny Edges", edged)
cv2.waitKey(0)
```



```
# Create copy of original image to draw all contours
img1 = image.copy()
cv2.drawContours(img1, cnts, -1, (0,255,0), 3)
cv2.imshow("4- All Contours", img1)
cv2.waitKey(0)
```




```
# Top 30 Contours
img2 = image.copy()
cv2.drawContours(img2, cnts, -1, (0,255,0), 3)
cv2.imshow("5- Top 30 Contours", img2)
cv2.waitKey(0)
```

```
# Drawing the selected contour on the original image
#print(NumberPlateCnt)
cv2.drawContours(image, [NumberPlateCnt], -1, (0,255,0), 3)
cv2.imshow("Final Image With Number Plate Detected", image)
cv2.waitKey(0)
```


```
Cropped_img_loc = 'Cropped Images-Text/7.png'
cv2.imshow("Cropped Image ", cv2.imread(Cropped_img_loc))
```

Output

```
Number is : HR26 BP3543
Out[6]: -1
```

Drawbacks of Vehicle Plate Algorithm

The following are the drawbacks of license plate Algorithm to use for electricity meter reading recognition:

- This Algorithm can also recognize/identify the static digits/letters, when the digits are dynamic and rotating (like in Electricity meter) this algorithm fails to recognize the digits
- This algorithm fails in recognizing the digits when the number plate has mud on it.

HD MR Algorithm

As the we saw that the license plate algorithm cannot recognize the rotating digits, we have used HD MR Algorithm, HD MR stands for Hausdorff distance for meter reading. This algorithm is trained to identify the rotating digits with a cutoff screen space of that digit in the meter. When the digit has moved forward for about 85% of the screen then we will consider that value as the next upcoming digit.
Recognizing the digits from the captured image requires 3 sub processes. We must preprocess the image to identify the location of numbers in the electricity meter and their rotation status. The three stages are:

i. Image Capture
ii. Preprocessing
iii. Number Recognition

Image Capture:
To make sure the reflection of light on the transparent lid of electricity meter from capturing the better-quality image to identify the digits without errors, we have a specific location to place the camera in front of the electricity meter. the camera position, distance, and angle from the lens to the object. The distance from the floor to the base of the camera is set to 29 cm, the distance from the floor to the base of the electrical meter is set to 22 cm, and, finally, the distance from camera lens to the meter is set to 5.5 cm.
Preprocessing:
The preprocessing is based on the hue, saturation, and value (HSV) color model. HSV has been used to locate car license plates. The method is advantageous in meter reading because of the combination of colors. It is possible to obtain a horizontal frequency histogram to determine the vertical bounds of the reading. After the analysis is completed, the procedure obtains a set of segmented numbers.
Number Recognition: To Recognize the numbers in the electricity meter we will use the following steps mentioned in the HD MR Algorithm:
Step 1: The step recognizes if a single fragmented frame represents a complete number or parts of 2
numbers when they are rotating. For this purpose, we have 2 portions of the frame, the upper part and the lower part. The upper part represents the percentage of the frame with the number that will be replaced after rotation. The lower part represents the percentage of the frame with the number that will replace the prior number in the reading. This kind of differentiation is possible because of the blank space area

between the numbers in rotation. The below picture Shows the different stages of a rotating number in electric meters.

Step 2: The procedure defines a threshold for a rotating number to be designated as 'complete'. For
this application, the threshold is set to 85% of the lower part. In other words, the system will only attempt to recognize a number when it has reached 85% visibility by the camera. This measure increases the HD recognition rate for individual numbers.
Step 3: Recognition is not attempted on partial numbers with a visibility under the

established threshold. The reading for this specific position is kept identical to the one previously stored. Hence, this position is not considered to have changed its value yet. By applying this strategy, the recognition time is reduced because the system only attempts to recognize an image when it surpasses the threshold.
Step 4: When all of the frames corresponding to a reading are recognized, the previous recognition value is used to evaluate the correctness of the actual reading. The actual reading must be greater than or at least equal to the last one stored. If this is not the case, the reading is set to the one previously stored.
Step 5: The actual reading is stored as the previous recognition; the system waits until a new image is
ready to be recognized and it starts again with step 1.

Conclusion

The HD MR algorithm has saved a lot of time and man power and also brought accuracy to the table. Study of this algorithm has helped us understand not only the image recognition of a dynamic image capture but also the underlying algorithm of vehicle plate recognition that helps in recognizing number plates in images that are captured when the vehicles are in motion. Given, time and sources, this study of HD MR Algorithm could have been a hands-on experience.

Bibliography

[1] Elihos, Alperen, Burak Balci, Bensu Alkan, and Yusuf Artan. "Deep Learning Based Segmentation Free License Plate Recognition Using Roadway Surveillance Camera Images." *Computer Vision and Pattern Recognition* (Computer Science > Computer Vision and Pattern Recognition), 2019.

[2] Elrefaei, Lamiaa. "Automatic Electricity Meter Reading Based on Image Processing." *Conference: IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies* . 2015.

[3] RODRIGUEZ, Miguel, Geovanni BERDUGO, Daladier JABBA, Maria CALLE, and Miguel JIMENO. "HD_MR: a new algorithm for number recognition in electrical meters." *Turkish Journal of Electrical Engineering & Computer Sciences* vol. 22 (2014): pp. 87 - 96.

[4] Xie, Fei, Ming Zhang, Jing Zhao, Jiquan Yang, Yijian Liu, and Xinyue Yuan. "A Robust License Plate Detection and Character Recognition Algorithm Based on a Combined Feature Extraction Model and BPNN." *Journal of Advanced Transportation* , 2018: 18.