Proceedings of the Fifth International Conference on Electronics, Communication and Aerospace Technology (ICECA 2021)
IEEE Xplore Part Number: CFP21J88-ART; ISBN: 978-1-6654-3524-6

Smart Parking Fee Detection Using Contour detection algorithm

[1]G.Shobana
Assistant Professor,
Dept of Information Science and Engineering,
Kumaraguru College of Technology,
Coimbatore
Shobana.g.cse@kct.ac.in

[3] D.Yamunathangam,
Assistant Professor,
Dept of Computer Science and Engineering,
Kumaraguru College of Technology,
Coimbatore
Yamunathangam.d.cse@kct.ac.in

[2] S.Balasubramanian, M. Ragul, U. Praven
UG scholar, Dept of CSE, Kumaraguru College of Technology, Coimbatore

***Abstract*— In today's world, many people are traveling in their own vehicles. The manual parking bills in public areas which leads to huge traffic or queue are formed for collecting the parking bills. Manual parking fee generation is a time consuming process, there is a possibility for loss of paper parking bills which leads to a lot of confusion, and sometimes there is lots of miscalculation in the parking fee. To reduce the time, the proposed model uses a vehicle number plate for the detection of parking fees. The automatic parking bill system in public areas through the vehicle number plate recognition will increase the convenience of collecting parking fees without the hassles of using paper bills. The system uses image processing that is Python OpenCV for localizing number plates for the billing system. It uses a Canny edge detection algorithm for localizing number plates from car images followed by character segmentation and uses a trained Python PyTesseract package is used for the recognition of characters in the number plate and generating a parking fee bill. As the result the system has reduced the interaction of humans and paper bills, time consumption is less. Cannon edge detection has improved the accuracy in the identification of the vehicle. The parking fee bill is generated at the time of exit of the vehicle from the parking area.**

Keywords*— *Python OpenCV, PyTesseract, Canny edge detection.

I. INTRODUCTION

The word automation sounded dreamy a couple of decades ago however in the current generation automation in every field is growing like a forest fire. We could experience automation almost daily in our day-to-day technology immense life. Automatic parking bill generation can be one of the revolutionary inventions in the transport field. To identify vehicles automatically we use character recognition and image processing. For the system to work. we need certain features like character recognition, character segmentation, image processing etc. The numbers can be recognize using Optical Image Recognition (OCR) algorithm etc. The image is converted to binary format. The objective of the project is to collect the necessary information from the image of the car and later use them to recognize the vehicle and produce the electronic bill for them. The field we are working on i.e., Automatic electronic bill generation is one of the growing research topics.

II. RELATED WORK

In this section, there are works done by various researchers using different methodologies and algorithms on number plate recognition. Brief description of some of them:

Chang, S.-L et al., had described the recognition of number plates from car images, they have used a Gaussian smoothing algorithm that helps to remove the noise. For detecting number plate location in the vehicle, they have used an algorithm based on edges called Sobel vertical edge detection. The algorithm is repeatedly used to detect the position of the number plate in the image. Once the number plate is detected the characters are segmented based on the gaps between the characters. They have used optical character recognition along with an artificial neural network is used for character recognition. Already Artificial Neural Network (ANN) is trained with a different class of characters based on the training already given the characters are recognized. It had an accuracy level of 82%.[1]

2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA) | 978-1-6654-3524-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICECA52323.2021.9675862

Du, S., Ibrahim, M., had suggested the detection of number plates using images that are captured from CCTV. So, basically from CCTV cameras, the images are captured, that images are used as input data. They have used mid filtering method for noise reduction in the image. The detection of a number plate is done using an algorithm called Sobel edge detection the same algorithm used in the previous paper. The character segmentation is done based on the vertical projection method which was a time-consuming and the probability of success was quite low. The characters were recognized using optical character recognition based on the number of endpoints comparing them with a 3x3 mask. Since it was based on the number of endpoints there was a lot of confusion in recognizing some characters example number 9 and number 6. Thus, it had a low accuracy level in recognizing the characters.[2]

Łubkowski, P., & Laskowski, D had suggested, Optical character recognition. It uses an artificial neural network as its method for recognizing characters. They have used a Multi-Layered preceptor Artificial Neural Network system (MLANN) for the recognition of characters. For classification purposes, they have used the Nearest Neighbour approach in the system. The MLANN system has three layers:

Input layer: It receives inputs in the form of segmented character images of accepted size.

Hidden layer: Trains the neural network regarding specific font styles in the process of character recognition.

Output layer: It is used to generate Unicode values for different characters which itself worked as a matching criterion.

As the result, this process had an accuracy of 96% in recognizing the characters.[3]

Björklund, T., et al., have suggested to used computer vision and image processing techniques to localize a license plate in an image. That includes morphological operations, thresholding, image gradients, bitwise operations, and contours for number plate detection in the image. This system works very fine in controlled and predictable environments. When there is lighting across input images and license plates are standardized such as dark characters on a light license plate background. They have used OpenCV, Tesseract, and PyTesseract to OCR number plates automatically. Since they have used PyTesseract in the system the computation was very fast in detecting the number plate in the image and recognition was also done fast. The accuracy of detecting the number plates and character recognizing was 95%.[4]

Panchal, T described the License plate detection using Harris corner and character segmentation by integrated approach from an image of Automatic number plate detection. First, the image of the car is taken as input. For preprocessing they have converted the RGB image to a Grayscale which enhances easy detection of the number plate in the image. For detecting the location of the number plate in the image they have used an algorithm called "Region props bounding box" in MATLAB. The bounding box is used to gauge the properties of the image region. To label the associated components, the region will extricate from the input image. Through vertical and horizontal scanning techniques the characters in the number plate are segmented based on pixels of the image. Two methods were proposed for character recognition

i)Template matching and

ii) Optical character recognition.

Both methods were able to recognition the process but the computational time, as well as the accuracy, were better in OCR compared to template matching.[5]

Azam. S.& Islam M. M had proposed to use IR sensor and MATLAB for detecting the number plate and for recognition of characters in the number plate.

The IR sensor is used to detect the vehicle and the signal is sent to MATLAB program using Arduino board which preprogrammed. The program captures the image from the video feed, and it detects the number plate from the image they have used morphological processing for image processing. For recognizing the characters, they have a database that has predefined fonts. When each character is segmented, every character for the recognition then the character is compared with a database with predefined fonts and then they are recognized. The proposed system can be able to recognize all kinds of fonts of characters.[6]

Yuan. Y, et al., had proposed the model by implementing Wpod-Net which is a pre-trained model to detect and extract License Plates from vehicle images. The system was able to detect the number plates of vehicles from different countries such as Germany, Vietnam, Japan, Thailand, Saudi, Russia, Korea, USA, India, China. Once the number plate is detected then characters are segmented in the number plate using OpenCV. Then for recognizing the

characters they have trained a Neural Network model for recognition.[7]

Davis.A. M., & Arunvinodh had processed the inputs acquired by the system are usually messy and have various sources. Inputs are given to an ML model (or neural network), to be standardized and cleaned. So, the colored images are converted to grayscale to reduce computation complexity. One important constraint that exists in some machine learning algorithms, such as convolution Neural networks (CNN), is the need to resize the images in your dataset to a unified dimension. Augmenting the existing dataset with perturbed versions of the existing images. Scaling, rotations and other affine transformations are typical.[8]

Xie, L., Ahmad, T et al., have used Yolo (You Only Look One) is a deep learning object detection architecture based on convolution neural networks (CNN), first, the image is processed by converting the RGB image to grayscale and then the grayscale image goes through binarization, then the background areas are blurred. The Segmentation and Recognition of characters are done using deep learning by providing predefined characters and the location of the characters in the number plates. This is discussed that this algorithm is efficient and optimized.[9]

Javale D et al. had suggested the optimal parking rate for the vehicle in the public area based on the parking space. Fees are calculated based on some parameters such as kind of day, current occupancy of the parking slot, and impact of the nearby parking area. Based on the previous history of occupancy of the vehicle in the parking area, the discount is provided through a machine learning algorithm. The parking system uses Raspberry Pi and LED sensors and an android/IOS mobile application system for the end-users.[10]

Prachu J. Patil had given a solution for the detection of an accident spotting system. The vibration sensor is placed in the vehicle for detection of the accident, the alert will send to the near traffic control room using the android application, Location of an accident is detected by the GPS module. The parking of the vehicle is also done GPS module, which helps to park the vehicle automatically in the vacant lane [13].

Suggest the smart parking of the vehicle. Which use sensors such as infrared, passive infrared, and ultrasonic sensors for identifying the vacant place in the parking area to park the vehicle. All the information about the parking is stored in the cloud it then connected to a mobile application. It will provide information regarding the availability of parking space. It helps the user to book the parking slots accordingly [11]. In the existing system, the character recognition from the number plate has less accuracy. In some papers, they used an IR sensor for detecting the number plate, which will lead to failure in monsoon season.

PROPOSED METHODOLOGY

The proposed system will capture the image of the vehicles and identify the number plate of the vehicles, and then the parking fee is detected automatically. The steps of the process are given (Fig 3.1).

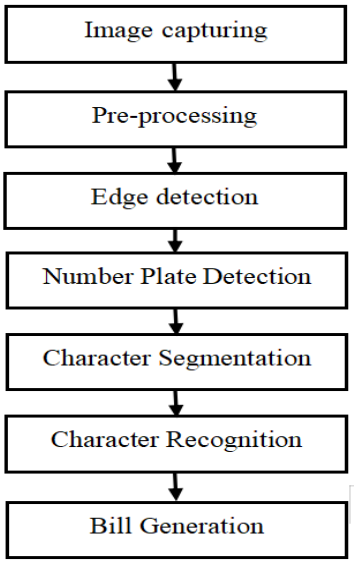

Fig 3.1 Architecture diagram

3.1 Proposed ANPR system

3.1.1 Image acquisition

The input image for pre-processing is taken from a camera in the parking counter in real-time. As we are using the dataset which contains the car image.

3.1.2 Image pre-processing

The images that were fed to the system as input may have noises so they must be pre-processed. As a first step, the image is re-sized to a constant resolution that is 620x480, as re-sizing helps to avoid problems with high-resolution images. Then the image in RGB format is converted to grayscale as given in Fig 3.2. Then the image is filtered with a Bilateral filter which will remove the irrelevant details in the image that is noise is removed by blurring out irrelevant details.

Fig 3.2- Grayscale image of the vehicle

3.1.3 Edge detection

After pre-processing the image, the next step is edge detection. The edge detection is we must identify the plate from vehicle images. The Canny edge detection method from OpenCV is an effective method for edge detection. A fixed range threshold value is set for the system the range is 30-200. The edges in the image which have an intensity gradient more than the minimum threshold and less than the maximum threshold are detected and are displayed.

3.1.3.1 Canny edge detector

It is a multistage algorithm that is widely used to detect edges in an image. A fixed range threshold value is set for the system the range is 30-200. The edges in the image which have an intensity gradient more than the minimum threshold and less than the maximum threshold are detected and displayed as shown in Fig 3.3.

Fig 3.3 Edge detection

3.1.4 Number plate detection

After edge detection now it is time to check out contours in the image. All the contours in the image are detected and then they are sorted in order from large to small. From that list of contours, only the top 10 are considered and the remaining are excluded. All the closed surfaces including the number plate will there in the top 10 list, now it is we need to filter out the number plate. This is now by looking out for rectangular contour by checking all the contours in the top 10 list in a loop. As a result of the loop, we will have our number plate detected. After the edge detection now, it is time to check out contours in the image [12].

3.1.4.1 Contour detection algorithm

A contour is defined as an outline representing or bounding the shape or form of an object. For which we use Contour Detection Algorithm which is fundamentally required for performing practical tasks, such as object recognition and scene understanding. Contour detection attempts to extract curves that represent object shapes from images. All the contours in the image are detected and then they are sorted in order from large to small to locate the number plate. From that list of contours, only the top 10 are considered and the remaining are excluded. All the closed surfaces including the number plate will there in the top 10 list, now it is we need to filter out the number plate [12]. This is now by looking out for rectangular contour by checking all the contours in the top 10 list in a loop. As a result of the loop, we will have our number plate detected as shown in Fig 3.4

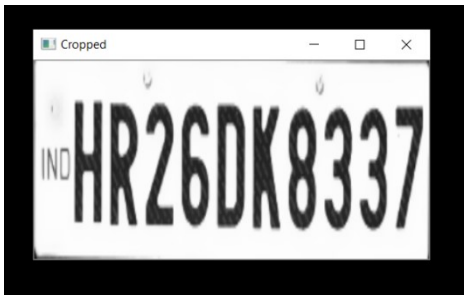

Fig 3.4 Number plate detection

3.1.5 Character segmentation

After detection of the number plate, the remaining parts of the image are not required so we mask the remaining parts of the image except for the number plate. Then the number plate alone is cropped out and it is saved as a new image. The image is used for character recognition. The cropped image is also converted grayscale and edge it for making smooth character recognition. Now based on the gaps between the characters they are segmented by cropping out each character. Within the plate, we search for obstacles with equal space between them based on that the characters are segmented by cropping each character.

3.1.6 Character recognition

The next step is recognizing the segmented characters from the number plate. To do that we have attached a pre-trained PyTesseract package to the system which is used for recognizing the characters from the number plate. Each segmented character is passed to the PyTesseract

package based on the training given to the package, it recognizes the characters by comparing it with the trained dataset. The number in the number plate is recognized and it is given as output as a result.

3.1.6.1 PyTesseract

PyTesseract is a built-in module of python language. It converts a 2D image to text form. The image is preprocessed like Binarisation and then the text is localized in the image and every word is detected. The recognition starts by trying to recognize each word then they are passed to an adaptive classifier which is a trained dataset here once again the word is recognized accurately, and the result is given out as text is shown in Fig 3.5

```
License Plate Recognition

Detected license plate Number is: ~HR26DK8337
```

Fig 3.5 Vehicle number as Text

3.1.7 Bill generation

Once the number is detected from the number plate the timer is set to start and it keeps on running. When the same car comes out of the parking lot again the same process takes place and a number plate is detected and the characters are recognized. The number recognized is already recorded then the timer of that car is stopped and then the total amount is calculated in terms of minutes. Based on the number of minutes the fee is calculated and a parking fee is generated. The parking offers are given to the regular customer. The vehicle with the same is number is parking daily, there is reduction the parking fees when compared to other customers. Based on this, parking fee also calculated for regular customer and automatically bill is generated for the customer.

IV. CONCLUSION

The benefits of automation in industries are quite remarkable especially in one of the frequently busy departments like parking. Electronic billing system performance is good and can be used commercially. In this project, we used character recognition algorithms to recognize the vehicles and provided an electronic billing system to provide the users with the accurate bill amount during their exit.

REFERENCES

[1] Chang, S.-L., Chen, L.-S., Chung, Y.-C., & Chen, S.-W. (2004, March). Automatic license plate recognition. IEEE Transactions on Intelligent Transportation Systems, 5(1), 42–53. doi:10.1109/TITS.2004.825086

[2] Du, S., Ibrahim, M., Shehata, M., & Badawy, W. (2013, February). Automatic License Plate Recognition (ALPR): A State-of-the-Art Review. IEEE Transactions on Circuits and Systems for Video Technology, 23(2), 311–325. doi:10.1109/TCSVT.2012.2203741

[3] Łubkowski, P., & Laskowski, D. (2017). Assessment of Quality of Identification of Data in Systems of Automatic Licence Plate Recognition. In J. Mikulski (Ed.), Smart Solutions in Today's Transport. TST. Communications in Computer and Information Science (Vol. 715). Cham: Springer. doi:10.1007/978-3-319-66251-0_39

[4] Björklund, T., Fiandrotti, A., Annarumma, M., Francini, G., & Magli, E. (2017). Automatic license plate recognition with convolutional neural networks trained on synthetic data. In 2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP), Luton (pp. 1-6). doi:10.1109/MMSP.2017.8122260

[5] Panchal, T., Patel, H., & Panchal, A. (2016). License plate detection using Harris corner and character segmentation by integrated approach from an image. Procedia Computer Science, 79, 419–425. doi:10.1016/j.procs.2016.03.054

[6] Azam, S., & Islam, M. M. (2016). Automatic license plate detection in hazardous condition. Journal of Visual Communication and Image Representation, 36, 172–186. doi: 10.1016/j.jvcir.2016.01.015

[7] Yuan, Y., Zou, W., Zhao, Y., Wang, X., Hu, X., & Komodakis, N. (2017, March). A robust and efficient approach to license plate detection. IEEE Transactions on Image Processing, 26(3), 1102–1114. doi:10.1109/TIP.2016.2631901 PMID:27893394

[8] Davis, A. M., & Arunvinodh, C. (2015, March). Automatic license plate detection using vertical edge detection method. In 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS) (pp. 1-6). IEEE. doi:10.1109/ICIIECS.2015.7193073

[9] Xie, L., Ahmad, T., Jin, L., Liu, Y., & Zhang, S. (2018, February). A New CNN-Based Method for Multi-Directional Car License Plate Detection. IEEE Transactions on Intelligent Transportation Systems, 19(2), 507–517. doi:10.1109/TITS.2017.2784093

[10]. Javale D., Pahade A., Singal R., Potdar A., Patil M. (2020), Smart Detection of Parking Rates and Determining the Occupancy. In: Bhalla S., Kwan P., Bedekar M., Phalnikar R., Sirsikar S. (eds) Proceeding of International Conference on Computational Science and Applications. Algorithms for Intelligent Systems. Springer, Singapore.

[11]. Abhirup Khanna, Rishi Anand (2016)," IoT based Smart Parking System" International Conference on Internet of Things and Applications (IOTA) pune

[12].Kranthi, K.Pranathi, A.Srisaila(2011), "Automatic Number Plate Recognition", International Journal of Advancements in Technology, ISSN 0976-4860

[13]. G. Vennila, Dr. D. Arivazhagan, Dr. R. Jayavadivel (2020), "An Analysis of Smart Car Parking Management System", International Journal of Scientific & Technology Research ISSN 2277-8616

[14] Patil, Prachu J., Ritika V. Zalke, Kalyani R. Tumasare, Bhavana A. Shiwankar, Shivani R. Singh, and Shailesh Sakhare(2021). "IoT Protocol for Accident Spotting with Medical Facility." Journal of Artificial Intelligence 3, no. 02 pp: 140-150.