**Mini Project Report on**



**Vehicle number plate detection and recognition in python**



**Submitted in partial fulfillment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Name: Mohd Owais Khan**  **University Roll No: 2018938**

***Under the Mentorship of***

**Ms. Meenakshi Maindola**

**Assistant Professor**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**January 2024**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“ Vehicle number plate detection and recognition in python”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Ms. Meenakshi Maindola, Assistant Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

**Name**: Mohd Owais Khan **University Roll no:** 2018938

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | 1-2 |
| Chapter 2 | Literature Survey | 3-4 |
| Chapter 3 | Methodology | 5-7 |
| Chapter 4 | Result and Discussion | 8 |
| Chapter 5 | Conclusion and Future Work | 9-10 |
|  | References | 11 |

**Chapter 1**

**Introduction**

Computer vision stands out as a prominent field within artificial intelligence, captivating our interest due to its connection with the fundamental human sense of sight. In particular, we are drawn to the significance of visual data and have chosen to focus our project on automatic registration plate recognition. This application not only aligns with the essence of human sight but also allows for the immediate extraction and display of license plate numbers through Optical Character Recognition (OCR) upon detection.

The given problem statement is to identify and extract license plate information from images.

The objective of our project is to develop a robust solution for identifying and extracting license plate information from images. License plate detection is a vital computer vision task with applications ranging from automatic toll collection and parking management to traffic monitoring and law enforcement. This technology finds utility in entrance checks, security systems, parking control, airport or harbor cargo management, road traffic control, and speed monitoring. Law enforcement agencies leverage license plate recognition systems to identify stolen vehicles, track suspects' movements, and enforce traffic regulations.

Furthermore, automated toll collection systems utilize license plate recognition to streamline the toll payment process. The chosen programming language for this endeavor is Python, complemented by its built-in deep learning and machine learning modules, including OpenCV, easyocr, imutils, numpy, and matplotlib. OpenCV, a key component in our toolkit, is a versatile library of programming functions designed primarily for real-time computer vision applications.

The methodology for vehicle identification involves a systematic approach, encompassing four main steps: pre-processing, license plate region extraction, character segmentation, and individual character recognition within the licensed number plate. This project seeks to contribute to the advancement of computer vision applications, particularly in the domain of license plate recognition, with practical implications across various sectors, including transportation, law enforcement, and security systems.

**Chapter 2**

**Literature Survey**

Python provides a robust framework for the efficient development of a vehicle number plate detection model, requiring minimal resources and effort. Several essential libraries contribute to the success of this project:

**1.** **OpenCV (Open-Source Computer Vision):**

OpenCV stands out as a widely-used computer vision library, offering a comprehensive set of tools for image and video processing. Its capabilities encompass image manipulation, feature detection, and object recognition, making it indispensable for vehicle number plate detection.

**2. NumPy:**

NumPy, a powerful numerical computing library, plays a vital role in handling arrays and matrices, which are foundational data structures in image processing. Its array manipulation functions enhance the efficiency of numerical operations.

**3. Imutils:**

Imutils simplifies the development process by providing a collection of convenient functions specifically designed for OpenCV. These utilities facilitate tasks such as image resizing, display, and other common operations in computer vision.

**4.** **Matplotlib:**

Matplotlib, a versatile plotting library, proves valuable for visualizing images and results during the developmental and testing stages of the number plate detection application.

For optimal character recognition (OCR) – a crucial aspect of vehicle number plate detection – the extracted license plate undergoes processing through an OCR algorithm. This algorithm recognizes and converts alphanumeric characters into machine-readable text.

To locate the actual position of the number plate, a series of image processing steps are performed. The process begins with grayscale conversion, followed by the application of filters to eliminate noise. Edge detection is then applied to identify edges within the image. OpenCV is employed to detect and extract contours, which are curves that connect points along the boundary of a shape with the same color or intensity.

In summary, the combination of these powerful Python libraries, along with dedicated image processing techniques, facilitates the creation of an efficient and accurate vehicle number plate detection model.

A diagram of a software development process

Description automatically generated

**Chapter 3**

**Methodology**

Our model operates through four primary steps, each playing a pivotal role in the detection and recognition of number plates:A diagram of a software system

Description automatically generated

The proposed model adeptly captures an image of the license plate, subsequently processed by a Python module for number plate extraction. This step showcases the License Plate Detection process (Localization), wherein both Grayscale and Canny images are generated.



Character Segmentation of the number plate holds paramount importance in the vehicle license plate detection system, as the accuracy of character recognition hinges entirely on effectivesegmentation. A close-up of a car

Description automatically generated

The image of the vehicle is acquired, with the goal of identifying the vehicle's number plate. The process involves the following key steps:

**1.** **Edge Filtering:**

Employing an edge filter proves instrumental in identifying boundaries within an image. This filter detects edges by pinpointing sharp variations in intensity gradients. Recognizing edges in the number plate is crucial as it streamlines data and eliminates extraneous information.

**2. Character Segmentation:**

In the realm of vehicle license plate detection systems, character segmentation stands out as a pivotal step. The efficacy of character recognition is contingent upon successful segmentation. This process involves isolating individual characters within the license plate.

**3. Character Recognition:**

Following segmentation, the subsequent step revolves around character recognition. The primary objective here is to convert the textual content within the image into recognizable characters. This recognition phase is integral to the overall process of identifying and interpreting the license plate information.

**Chapter 4**

**Result and Discussion**

We have successfully created and tested our model with our image.

**Output:**

A close up of a car

Description automatically generated

The results underscore the system's reliability in diverse scenarios, emphasizing its potential impact on enhancing traffic management and security.

A license plate on a red car

Description automatically generated

**Chapter 5**

**Conclusion and Future Work**

The proposed model adeptly captures an image of the license plate, subsequently processed by a Python module for number plate extraction. This step showcases the License Plate Detection process (Localization), wherein both Grayscale and Canny images are generated. Character Segmentation of the number plate holds paramount importance in the vehicle license plate detection system, as the accuracy of character recognition hinges entirely on effective segmentation.

It can be inferred that the aforementioned model holds immense utility in enumerating the individuals present in an image. The versatility of this model extends to applications in social and political events, household surveillance systems, military operations, and healthcare services. Given the escalating global population, the relevance and applications of this model are anticipated to burgeon.

To augment its capabilities, the model can be refined to seamlessly process real-time videos, providing a dynamic means to quantify the number of individuals present. Additionally, there is potential for the model's enhancement to categorize individuals based on gender and age groups, thereby expanding its utility in healthcare and law enforcement contexts for demographic profiling.

From a career standpoint, proficiency in deploying such a model proves invaluable. Beyond contributing to a professional portfolio with substantial and intricate projects, it affords hands-on experience with Python libraries associated with machine learning and deep learning. In the current competitive societal landscape, such skills are a distinct advantage.

Furthermore, avenues for improvement exist by incorporating alternative approaches, such as counting by detection, counting by clustering, and counting by regression. These strategies can further refine the model's accuracy and broaden its scope of applications. As technology continues to evolve, the continued development and refinement of such models are poised to make significant contributions across various domains.

**References**

**1. Marsden, M., McGuinness, K., Little, S., & O'Connor, N. (2017).** "Dense crowd counting from still images with convolutional neural networks." In Proceedings of the 8th ACM on Multimedia Systems Conference (pp. 99-104). ACM.

(https://dl.acm.org/doi/abs/10.1145/3083187.3083200)

**2. Sindagi, V. A., & Patel, V. M. (2017)** "Generating high-quality crowd density maps using contextual pyramid CNNs." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 5225-5234).

(http://openaccess.thecvf.com/content\_cvpr\_2017/html/Sindagi\_Generating\_High-Quality\_Crowd\_CVPR\_2017\_paper.html)

**3. Zhang, C., Li, H., Wang, X., & Yang, X. (2015)** "Cross-scene crowd counting via deep convolutional neural networks." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 833-841).

(https://ieeexplore.ieee.org/document/7298685)

**4. Liu, S., Liu, C., & Shao, J. (2019).** "Crowd counting via scale-adaptive convolutional neural network." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 11141-11150).(http://openaccess.thecvf.com/content\_CVPR\_2019/html/Liu\_Crowd\_Counting\_via\_CVPR\_2019\_paper.html)

**5.** **Sindagi, V. A., & Patel, V. M. (2019).** "Wide-area crowd counting: Dataset and algorithms." IEEE Transactions on Circuits and Systems for Video Technology, 29(5), 1427-1441. (https://ieeexplore.ieee.org/document/8644105)