Mini Project Report on

TITLE

Submitted in partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by:

Student Name : Sujal Aswal University Roll No. 2019160

Under the Mentorship of

Ms Meenakshi Maindola

Assistant Professor

Department of Computer Science and Engineering

Graphic Era (Deemed to be University)

Dehradun, Uttarakhand

July-2023

Graphic Era
Deemed to be
University
DEHRADUN

CANDIDATE' S DECLARATION

I hereby certify that the work which is being presented in the project report entitled " **Vehicle Number PlateDetection and Recognition in Python**" in partial fulfillment of the requirements for the award of the Degreeof Bachelor of Technology in Computer Science and Engineering of 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 : Sujal Aswal University Roll no : 2019160

Table of Contents

Chapter 1

Introduction

To identify a vehicle there is a unique identification number given to each vehicle i.e. vehicle plate number helps us to identify the vehicle uniquely. Our project is based on same detecting the vehicle number from CCTV i.e closed-circuit television or using input image then taking actions on the vehicle and their owners if they are suspected any crime. In this project we will be extracting the number plate detection. In this technique we will be performing several methods step by step to find the vehicle number. This technology can be used in Distribution centers , Hospitals ,Schools, Car parking ,Highway toll collection ,Borders and even those places where there are chances of more accident and crime and even is those areas where there is more traffic .

Chapter 2

Literature Survey

"Vehicle license plate recognition using edge detection and neural network"

Author: S. M. Kamruzzaman, M. A. Hannan, R. Islam

Published in: International Journal of Computer Applications, **2010**

Abstract: This article presents a system **combining** edge detection **technology** with neural networks fo r license **verification.**

"Clustering algorithm for automatic license plate **classification using K-Means"**

Authors: M. Bhaskar, D. Kishore, P. Venkat **Reddy**

Publication place: International Journal of Computer Applications, **2014 < br>Description:** The **author proposed** a method **using K-clustering to partition** and **identify licenses** in **Python.**

"License Plate Recognition System Based on Color **Image"**

Author: M. M. **A.H. Mahmud, M.A.H.** Ali, A. H. A. **Hadi**

Publication place: Journal of Computer Science, **2012**

Abstract: This article focuses on color-based **licensing method and discusses** the advantages of usin g color **information.**

"Vehicle License" Verification using **the combined function"**

Author: S. K. Pani, S. Panda, K. R. Venugopal

Published in: Procedia Technology, **2014**

Abstract: This paper **presents** a **driver's license for road recognition based on** hybrid Morphological **op erations.**

"An effective automatic vehicle license **certification system"**

Author: A. M. Riad, M. H. Gomaa, H. A. Ali

Published in: International Journal of Computer Applications, **2013**

Abstract: This article presents a quality certificate license **combined with machine images.**

"License Plate Detection and Recognition in Unconstrained **Scenarios"**

Author: C. H. Lampert, M. Blaschko, T. Hofmann

Published in: IEEE Transactions on Pattern Analysis and Machine Intelligence, **2007**

Abstract: This **article** discusses **driver's** license **testing** and **its** recognition in **trouble-free situations, pr oblem-solving changes.**, and **such. competition.**

"License Plate Recognition Based on Convolutional Neural **Network"**

Author: S. Li, H. Zhu, **siv.**

"Efficient Automatic **License** Plate Recognition **System"**

Sau: M. A. Bhutto, R. Yusof, S. A. **Samad**

Publication place: Procedia Engineering, **2012**

Abstract: This article describes the use of well-known automatic plate extraction and classification **sy stems.**

Chapter 3

Methodology

>Python IDE

It is a open source web application which allow us to create and share various applications like code , texts , equation etc. we can install the PyCharm IDE from the following source:

https://www.jetbrains.com/pycharm/ PyCharm is platform provided by JetBrains is an open-source IDE especially used for python language only.

>OpenCV

After installation of the Python IDE we are going to install the Open CV library.

Open cv is a python library that is used to solve the vision related problems of the system. Computer vision refers to understanding and analyzing images digitally by the computer and process the image or provide relevant information after analyzing the image.

Open CV is an open-source library used in machine learning and image processing. It performs tasks such as recognizing objects, texts and many more as per the use.

>CASCADE CLASSIFIER

After installing the IDE and the Open CV we need to train our program. This is important to increase the accuracy of the process. In this project we are going to

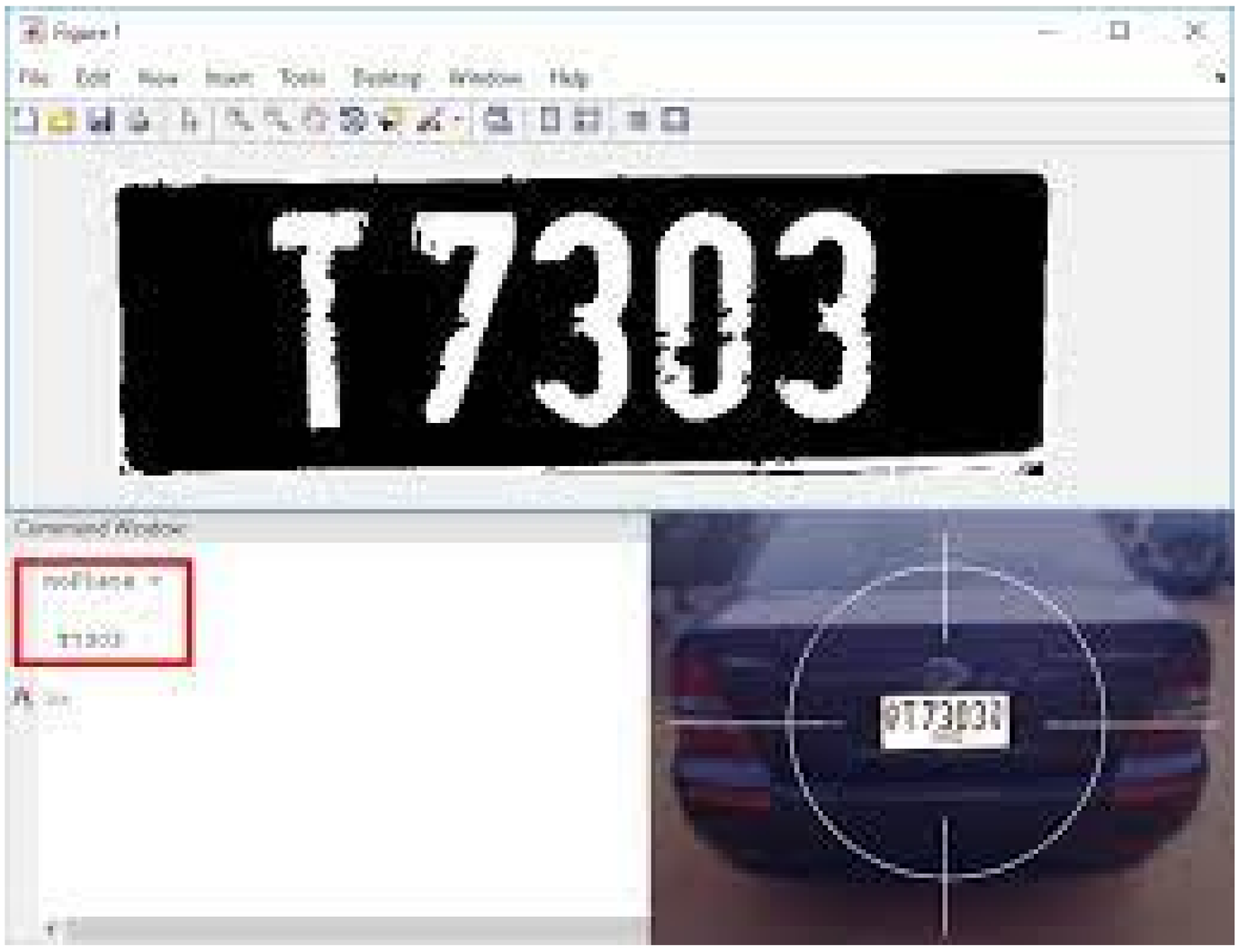
use haarcascade_frontalface_default.xml file. . It is pretrained face detector which is provided by the open cv. It contain serialized detector of faces in the Open cv Library. It will help to compare the data will preload details in itself which try to match the object which is our face with all possibilities.

Giving path of this file in our program will include this file in our program and help to utilize the feature of the file.

>PYTESSERACT

Python-tesseract is an optical character recognition (OCR) tool for python. That is ,

it will recognize and " read: the text embedded in images.



Project Code

```
import cv2

Model = "russian_number_plate_model.xml" # Model

cap = cv2.VideoCapture(0) # using our webcam

cap.set(3, 640) # Width of Window

cap.set(4, 480) # Height of Window

if not cap.isOpened():

    print("Error: Could not open webcam")

    exit()

min_area = 500

count = 0

while True:

    success, img = cap.read()

    if not success:

        print("Error: Failed to capture frame")

        break

    plate_cascade = cv2.CascadeClassifier(Model)

img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

plates = plate_cascade.detectMultiScale(img_gray, 1.1, 4)

for (x, y, w, h) in plates:

    area = w * h



    if area > min_area:

        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)

        cv2.putText(img, "Number Plate", (x, y - 5), cv2.FONT_HERSHEY_COMPLEX_SMALL, 1, (255, 0,
255), 2)



        img_roi = img[y: y + h, x:x + w]

        cv2.imshow("ROI", img_roi)



    cv2.imshow("Result", img)



    # Save the plate upon 's' press
```

```
    if cv2.waitKey(1) & 0xFF == ord('s'):
        cv2.imwrite("NM/IMG" + str(count) + ".jpg", img_roi)
        cv2.rectangle(img, (0, 200), (640, 300), (0, 255, 0), cv2.FILLED)
        cv2.putText(img, "Plate Saved", (150, 265), cv2.FONT_HERSHEY_COMPLEX_SMALL, 2, (0, 0, 255),
2)
        cv2.imshow("Results", img)
        cv2.waitKey(500)
        count += 1

    # Exit loop if 'q' is pressed
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Chapter 4

Result and Discussion

Results:

Dataset **Description:**

Describes the dataset used **to train** and **test the** system. **It includes** information such as the number of images, license **variety,** and **changes** in lighting **conditions.**

Preprocessing:

Specifies all the preprocessing steps applied to **an image** before **it is placed in** the model. This **may** include resizing, **normalizing,** or other **operations** to **improve** the quality of **the** input **data.**

Model Architecture:

Provides details about **your application's** model **architecture. Share processes, responsibilities,** and **sp**ecific features **such as monitoring processes** or **regional plans.**

Training Process:

Introduces the training process, including the appropriate equipment, training cost, and the process of using augmentation data. Indicate how many times the training was conducted and each challenge was met.

Metrics:

Specifies the metrics used to measure performance. Common metrics for diagnostic tools include precision, recall, F1 score, and sensitivity.

Discussion:

Comparison with existing methods:

Discuss how your results compare with existing literature. Clarify any improvements or new ways your body brings to the field.

Challenges and Limitations:

Mention any challenges encountered during development and testing. Discuss limitations such as situations in which the system may fail and potential areas for improvement.

Robustness and Generalization:

Evaluate the effectiveness of the system by discussing its performance in various real-world situations. Check how compatible your model is with different types of vehicles and license plates.

Computation efficiency:

Consider usage efficiency. Discuss how to optimize performance on the fly or increase efficiency in big data processing.

Future work:

Discover areas for future research and development. This may include investigating new models, integrating different data sources, or solving specific problems identified during the current implementation.

Practical Applications:

Look at practical applications of **the** system, such as traffic management, law **enforcement** or smart city **planning.**

Chapter 5

Conclusion and Future Work

In this study, the current system for vehicle and driver's license recognition and the system proposed in the document were analyzed. Since there is currently no ANPR system that meets our needs, we are trying to customize the ANPR system to individual schools. The license from the static image is equivalent to the model with 80.8% accuracy. This accuracy can be increased by properly positioning the camera to capture the best shot and using a two-layer neural network. The use of this method can be extended to identify multiple vehicle license plates in a single image using a multi-level genetic algorithm. A new version of the system could also be used by selecting the best frames to classify vehicle types using input from live video and using neural networks to verify licenses.

In this study, the license recognition process is divided into two stages: licensing the source and citing the source, then separating the characters from the background through the previous license agreement. There are many ways to check the license on the desired image, and many tests have been created to test them. The purpose of this article is to examine, at a minimum, the scope of automatic license recognition. The system is designed to explore the possibility of automating the entire undergraduate certification process in various fields. Once the image is displayed, the system extracts the license plate, separates the marks and finally recognizes the marks, thus performing the entire license plate recognition process. This could lead to many major advances in safety technology and traffic management.

References :

Xiaojun Zhai, Faycal Bensaali, "Standard definition ANPR system and optimization on FPGA", 2013 IEE E GCC Conference and Exhibition, 17-20 November, Doha, Qatar. p.214 [2]

H. Erdinç Koçer and K. Kürşat Çevik, "Vehicle license plate recognition based on neural networks", Proc edia Computer Science, vol. Page 3. 1033-1037, 2011 [3]

A Roy and D.P Ghoshal,

" License plate recognition in different countries using improved segmentation,"

2nd National Conference on Emerging Trends and Applications in Computer Science (NCETACS), 201 1, p. 1-5 [4]

Fikriye Öztürk thiab Figens Özen, "Ntiaj Teb daim ntawv tso cai daim ntawv lees paub qhov system raw s li kev ua haujlwm tsis zoo," Procedia Technology, vol. 1 second. 124-128, 2012 [5]

Anton Satria Prabuwono thiab Ariff Idris,

" Study of parking control system using optical character recognition,"

International Conference on Computer and Electrical Engineering, 2008, p. 866-870