INDUSTRIAL TRAINING REPORT

on

RUSSIAN LICENSE NUMBER PLATE DETECTOR

Submitted in partial fulfillment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY

Submitted by

Ishaan Bhugra
(35215603120)
to
Dr. Pinki Nayak, Associate Professor

Department of Information Technology
Dr. Akhilesh Das Gupta Institute of Technology & Management
(Guru Gobind Singh Indraprastha University, Dwarka, Delhi.) New Delhi -110053.
JANUARY 2023

CERTIFICATE:

ACHIEVEMENT

Allsoft Solutions
An ISO 9001 : 2008 CERTIFIED COMPANY

PROJECT COMPLETION CERTIFICATE

In recognition of the commitment to achieve professional excellence this is to certify that Ms./Mr.

ISHAAN BHUGRA

has successfully completed an Industry-oriented project.

Project Name	Russian License Number Plate Detector
Technologies Used	Artificial Intelligence – PYCHARM
Reference No.	CE/2022/F/AUG/0076
Training Date	2nd August, 2022 – 17th September, 2022
Training Duration	90 Hours
Training Location	Dr. Akhilesh Das Gupta Institute of Technology & Management

Program Co-ordinator
Industry/Academic Alliance

Business Partner IBM

Director
Training and Development
Allsoft Solutions and Services

BIG DATA - ANALYTICS | IoT | ORACLE | J2EE | PHP | CLOUD COMPUTING

ACKNOWLEDGEMENT

I would like to acknowledge the contributions of the following persons, without whose help and guidance this report would not have been completed.

I acknowledge the counsel and support of our project guide **Manish Sharma, Trainer, IBM** with respect and gratitude, whose expertise, guidance, support, encouragement, and enthusiasm has made this report possible. Their feedback vastly improved the quality of this report and provided an enthralling experience. I/We are indeed proud and fortunate to be supervised by him.

We are thankful to, **H.O.D. IT department, Dr. Akhilesh Das Gupta Institute of Technology & Management, New Delhi** for his/her constant encouragement, valuable suggestions and moral support and blessings.

I are immensely thankful to our esteemed, **Prof. (Dr.) Sanjay Kumar, Director, Dr. Akhilesh Das Gupta Institute of Technology & Management, New Delhi** for his never-ending motivation and support.

I shall ever remain indebted to**, Project Coordinators IT department** and faculty and staff members of Dr. Akhilesh Das Gupta Institute of Technology & Management, New Delhi.

Finally, yet importantly, I/We would like to express our heartfelt thanks to God, our beloved parents for their blessings, our friends/classmates for their help and wishes for the successful completion of this project.

ISHAAN BHUGRA

(35215603120)

ABSTRACT

Our project name is **Russian License Number Plate Detector**. It is one of the necessary systems designed to detect the vehicle number plate. In today's world with the increasing number of vehicle day by day it's not possible to manually keep a record of the entire vehicle. With the development of this system, it becomes easy to keep a record and use it whenever required. The main objective here is to design an efficient automatic vehicle identification system by using vehicle number plate. The system first would capture the vehicles image as soon as the vehicle reaches the security checking area. The captured images are then extracted by using the segmentation process. Optical character recognition is used to identify the characters. The obtained data is then compared with the data stored in their database. The system is implemented and simulated on Pycharm and performance is tested on real images. This type of system is widely used in Traffic control areas, tolling, parking area, etc. This system is mainly designed for the purpose of security system. However, the detection of the dynamic or moving object is a challenging part. In areas wherever automobile parking space is taken by a specific vehicle, the incorrectly parked vehicles are recognized. It is noted that the license plates of the vehicles are found in several forms, size and conjointly they're completely different in color. This makes the detection of the number plate of a vehicle, the foremost fascinating and difficult analysis topic. Number plate detection is useful in finding purloined cars, automobile parking management system and identification of the vehicle in traffic. The number plate decoded will be used further for identification, matching and documentation purpose of vehicle details.

This approach is simplified to identify the rectangular number plate using the frame capturing approach. After the detection of the plate, the program displays a box around the number plate. The main focus is to locate the number plate correctly on the moving vehicle.

TABLE OF CONTENTS

List of Figures

CHAPTER 1

1.1 INTRODUCTION

Number plates are utilized as distinguishing proof of vehicles everywhere throughout the countries. The number plate recognition system uses a picture handling technique for perceiving automobiles by their number plates. Number plate recognition systems are utilized with the point of viable movement control and security applications like access control to limited regions and pursue wanted vehicles. Experimentation of number plate recognition has been led for quite a long while; it's as yet a troublesome task. Number plate identification system investigates a data picture to distinguish some nearby fixes containing number plate. Since a plate can exist anyplace in a photo with fluctuated sizes, it's difficult to inspect every pixel of the picture to discover it. At the point when a vehicle enters an input gate, the number plate can naturally be detected at the entrance point and put away in the database. The Number Plate Recognition (NPR) system for Indian number plate is troublesomely contrasted with the foreign number plate as there's no standard took after for the proportion or ratio of the number plate size. The recognition task is difficult because of the nature of the light, which causes the image acquisition difficult. In NPR system photo-detection approach is used that includes acquiring a photo of the vehicle, extracting the region of interest, and character segmentation and extraction. It is troublesome to locate the bounding area or edge of the number plate from the input vehicle image in the open-air scene because of the shades of characters of the number plate and background of the same. The gradients of the original picture are modified to discover applicant number plate region. There are calculations that depend on a combo of morphological activity, division, segmentation, and canny edge identifier. Number plate location detection comprises of steps like as Edge Detection, a Morphological task like expansion and disintegration, smoothing, and division of characters and recognition of plate number. With the increasing number of vehicle in today’s world it’s not possible to manually keep a record of the entire vehicle. There need to be a man standing 24*7 to note down the number. It’s a time consuming process and require manpower. Furthermore the data stored manually is not readable after a long time. So to overcome all these limitations here we tried to develop a system which would automatically detect the number plate and store it in its database. Later on when the information is required one can get it and use it. This process also helps to get the correct result compared to manually one. The process of working

involves that as soon as the vehicle enters the secured area the system automatically captures the images and stores it. The processing of the image is done through the software stored in the system. If the vehicle matches the already stored information then it's allowed to pass the gate. And if the vehicle is not recognized or if its marked in the blocked list then it's not allowed to cross the gate and further checking process are followed.

1.2 INTRODUCTION TO PYTHON

Python is an interpreted **high-level general-purpose programming language**. Its design philosophy emphasizes code readability with its use of significant indentation.

Its language constructs as well as **its object oriented** approach aim to help programmers write clear, logical code for small and large-scale projects. Python is **dynamically-typed** and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a cycle-detecting garbage collection system (in addition to reference counting).

Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible. Python 2 was discontinued with version 2.7.18 in 2020. Python consistently ranks as one of the most popular programming languages.

1.3 HISTORY OF PYTHON:

Python was conceived in the late 1980s by **Guido van Rossum** at **Centrum Wiskunde & Informatica (CWI)** in the **Netherlands** as a successor to the ABC programming language, which was inspired by SETL, capable of exception handling and interfacing with the Amoeba operating system.

Its implementation began in December 1989. Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's "benevolent dictator for life", a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. In January 2019, active Python core developers elected a five-member "Steering Council" to lead the project.

1.4 INTRODUCTION TO PYCHARM?

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains (formerly known as IntelliJ). It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda.

PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License, and there is also Professional Edition with extra features – released under a proprietary license.

1.5 PROJECT

1.5.1 Russian License Number Plate Detector

Russian License plate detection is identifying the part of the car that is predicted to be the number plate. Recognition is identifying the values that make up the license plate. License plate detection and recognition is the technology that uses computer vision to detect and recognize a license plate from an input image of a car.

1.5.2 Objective:

In general, video surveillance systems are utilized for monitoring and security purposes. However, one difficult aspect of video surveillance is the detection of moving objects. Human activity identification and monitoring has grown more feasible in recent

years, thanks to the falling prices for high-quality video surveillance systems. As a result, automated systems have been created to perform a variety of detection duties, but the work of identifying unlawfully parked vehicles has generally been delegated to surveillance system operators. The most fascinating and difficult research topic over the past few years has been the identification of RUSSIAN automobiles by their license plates.

This approach is simplified to identify the rectangular number plate using the frame capturing approach. After the detection of the plate, the program displays a box around the number plate. The main focus is to locate the number plate correctly on the moving vehicle.

1.5.3 Working:

The development of the Russian License Number Plate Recognition System is being implemented using Pycharm by using following listed out process:

1. Firstly, the code is run and it shows the server.
2. After going to server it asks for the input , the input can be taken as a video or an image .
3. Then the input will be uploaded.
4. After the input is uploaded, immediately it is directed to the result page.
5. The result page shows the output of the license number plate as marking it edges.

CHAPTER 2
METHODOLOGY

2.1 STEPS INVOLVED

For software to detect and recognize a license plate, it undergoes three major processes.

- **Taking an image of a car as input** - The program takes in the input of the car in which the license plate is to be detected.
- **Processing the input** - The image taken as the input undergoes processing to detect the part of the car that is the license plate.
- **Recognizing the number plate** - The values of the detected license plate are extracted from the number plate image.

2.2 Software Model:

The first and the most important part in this process is the software model. The software model uses the image processing technology. The programs are implemented in Pycharm. The algorithm is divided into following parts: Capture image, Pre-processing and Plate region extraction. The flow chart of license plate recognition system implementation in this work is shown in the following figure. There are various steps in this approach and these are implementation in Pycharm.

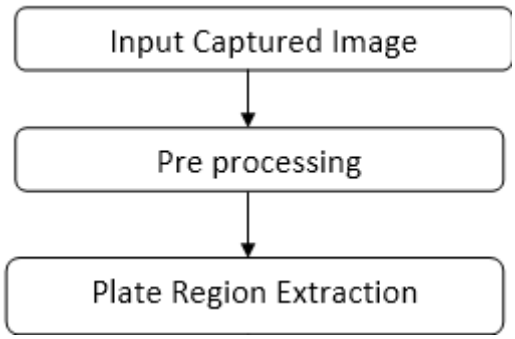

Fig 2.1 Software Model

2.3 Libraries Used:

- **Tkinter**
- **OpenCV**
- **Haar Cascade XML File**
- **Numpy**
- **Flask**

2.3.1 Tkinter

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method.

It is a standard Python interface to the Tk GUI toolkit shipped with Python.

Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

2.3.2 OpenCV

OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library.

It mainly focuses on image processing, video capture and analysis, including features like face detection and object detection, and it helps to provide a common infrastructure for computer vision applications.

We install the Python version of OpenCV (in your virtual environment) with the following command:

Pip install opencv-python

2.3.3 Haar Cascade XML File

Besides installing the OpenCV library, another important thing to retrieve is the **Haar Cascade** XML file.

Let’s first talk about the theory behind Haar Cascades since it is an important concept.

In 2001, Paul Viola and Michael Jones came up with the object_detection technique using Haar feature-based cascade classifiers.

It is a machine learning based approach (involving AdaBoost) where a cascade function is trained from many positive and negative images.

It extracts numerical values for features (e.g. edges, lines) efficiently with the concept of **integral image** (or summed-area table), which trumps the default computationally-heavy way of subtracting sums of pixels across multiple regions of an entire image.

In addition, it uses the '**Cascade of Classifiers**'.

This means that instead of applying hundreds of classifiers for the many features within the image at one go (which is very inefficient), the classifiers are applied one by one.

OpenCV actually comes with pre-trained XML files of various Haar Cascades, where each XML file contains the feature set. We will be using the Haar Cascade XML file containing the features for Russian car plates.

2.3.4 Numpy

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.NumPy was created in 2005 by Travis Oliphant.

It is an open source project and you can use it freely. NumPy stands for Numerical Python.

2.3.5 Flask

Flask is a web framework, it's a Python module that lets you develop web applications easily. It's has a small and easy-to-extend core: it's a micro framework that doesn't include an ORM (Object Relational Manager) or such features.

It does have many cool features like url routing, template engine. It is a WSGI web app framework.

2.4 GUI

The **graphical user interface** is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicator such as primary notation, instead of text-based user interfaces, typed command labels or text navigation. GUIs were introduced in reaction to the perceived steep learning curve of command-line interfaces (CLIs), which require commands to be typed on a computer keyboard.

CHAPTER 3

3.1 RESULT ANALYSIS

- ✓ The program works successfully according to our requirements.
- ✓ The picture of vehicle number plate is taken with the image capturing system and the license number of the vehicle is marked.
- ✓ It simply places a blue rectangular box anywhere above the area which seems like a Russian Number Plate of the cars.
- ✓ The system can even detect the Russian Number Plates from a distance as shown in the snapshots below.

3.2 SNAPSHOTS OF PROJECT

Fig 3.1 Snapshot-1

Fig 3.2 Snapshot-2

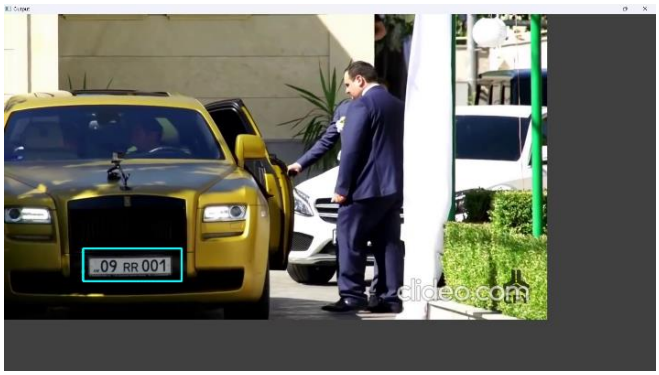

Fig 3.3 Snapshot-3

Fig 3.4 Snapshot-4

Fig 3.5 Snapshot-5

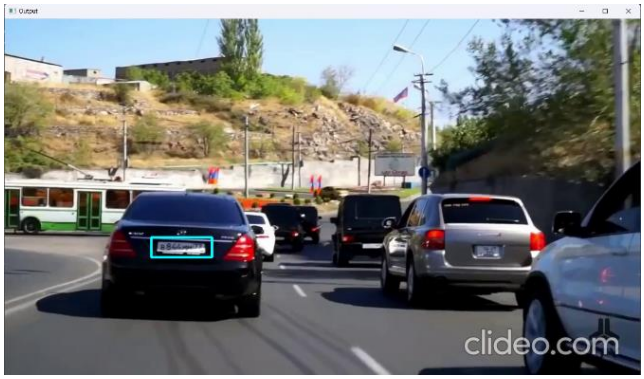

Fig 3.6 Snapshot-6

CHAPTER 4

MERITS, DEMERITS AND APPLICATIONS

4.1 Advantages:

- Improving road safety.
- Reduces crimes.
- Gives officers better information to work with.
- Deterring terrorism.
- Pre-paid members can be easily differentiated from non-members.
- Giving a greater police presence.

4.2 Disadvantages:

- Firstly, the images of the number plate or of any object which is taken by using the optical character reader technology may get blurred mainly due to the reason of motion blurring for which the picture seems to be hazy when uploaded in the database.
- Secondly, the technology often uses low-resolution images for which the images are not actually visible properly in every case.

4.3 Applications:

- Parking.
- Access Control.
- Motorway Road Tolling.
- Border Control.
- Journey Time Measurement.
- Law Enforcement.

CHAPTER 5
CONCLUSIONS AND FUTURE SCOPE

5.1 CONCLUSION:

After so much effort put in, I successfully created a program that detects and recognizes license plates. In Russian License Plate Detector, the picture of vehicle number plate is taken with the image capturing system and the license number of the vehicle is marked. This project serves as a stepping stone for larger scale (and more advanced) computer vision projects, such as bulk extraction of car license plate text from large image quantities and applying these concepts on video files or live feed. I would like to extend my gratitude towards:

- IBM
- Mr. Manish Sharma(Trainer)

This thesis explains different recognition methodologies, their advantages and drawbacks and gives the best of all those to opt for a user friendly, efficient system that works in any climatic conditions unaffected. That system should not be effected by the factors like speed, light.

5.2 FUTURE SCOPE:

1) The number plate recognition system can be enhanced to record all types of number plates whatever their size and font type may be.

2) The working of this model can be made faster as the technique which are implemented at this time are slow they usually take around 2 second to provide complete result.

3) Could detect the fake Number plate and take the image of the driver and store it in the Database.

4) Could detect the vehicle which run fast if they are breaking the traffic rules than we could put fine on the vehicle driver.

REFERENCES

- https://www.marketsandmarkets.com/Market-Reports/anpr-system-market-140920103.html
- https://opensource.google/projects/tesseract
- https://www.linuxjournal.com/article/9676
- https://pdfs.semanticscholar.org/bdca/d2b56e3a38ef543f6fb0a602deb5f453493b.pdf?_ga=2.28647292.1514298175.1598968225-502553827.1598968225
- https://pdfs.semanticscholar.org/4d31/46d2b4bf23558ec0baf93506be5b96437fc2.pdf
- https://www.researchgate.net/publication/299858935_Proposal_for_Automatic_License_and_Number_Plate_Recognition_System_for_Vehicle_Identification
- https://opencv.org/#
- https://pypi.org/project/Pillow/
- https://www.python.org/about/gettingstarted/
- https://numpy.org/learn

APPENDIX

(Coding part of the project)

SOURCE CODE (main.py):

```
from flask import Flask, render_template , request
from flask_wtf import FlaskForm
from wtforms import FileField, SubmitField
from werkzeug.utils import secure_filename
import os
from wtforms.validators import InputRequired
from pathlib import Path
import tempfile
import cv2



app = Flask(__name__)
app.config['SECRET_KEY'] = 'supersecretkey'
app.config['UPLOAD_FOLDER'] = 'static/files'

class UploadFileForm(FlaskForm):
    file = FileField("File", validators=[InputRequired()])
    submit = SubmitField("Upload File")

@app.route('/', methods=['GET',"POST"])
# @app.route('/home', methods=['GET',"POST"])
def home():
   form = UploadFileForm()
   if form.validate_on_submit():
      file = form.file.data # First grab the file

file.save(os.path.join(os.path.abspath(os.path.dirname(__file__)),app.config['UPLOAD_FOLDER'],secure_filename(file.filename))) # Then save the file
```

```
  return render_template('index.html', form=form)

@app.route('/result', methods=['GET','POST'])
def result():
    if request.method=='POST':
        uploaded_file = request.files['file']
        with tempfile.TemporaryDirectory() as td:
            temp_filename = Path(td) / 'uploaded_video'
            uploaded_file.save(uploaded_file)
            vidcap = cv2.VideoCapture(str(uploaded_file))
    result_get="resultlicense"
    return render_template('result.html' , result_get=result_get)

def process():
    vidcap = cv2.VideoCapture('uploaded_file')
    return vidcap;
    return jpeg.tobytes()

if __name__ == '__main__':
    app.run(debug=True)
```

SOURCE CODE (project.py):

project.py

```python
from tkinter import Frame
import cv2
import numpy as np


capture = cv2.VideoCapture("projectvideo.mp4")
dataset = cv2.CascadeClassifier('haarcascade_russian_plate_number.xml')


while True:

    ret,frame = capture.read()

    plates = dataset.detectMultiScale(frame,1.2)
    print(plates)
    for x,y,w,h in plates:
        cv2.rectangle(frame, (x,y), (x+w, y+h), (255,255,0), 4)
        break
    ret,jpeg=cv2.imencode('.jpg',frame)
    cv2.imshow('Output',frame)
    k = cv2.waitKey(5)
    if k == 27:
        break


capture.release()
cv2.destroyAllWindows()

```

****