

AUTOMATIC NUMBER PLATE

RECOGNITION

- **Overview of the Project:**

Number Plate Recognition System is an image processing technology which uses number (license) plate to identify the vehicle. The objective is to build a web application that can efficient automatic authorized vehicle identification system by using the vehicle number plate. Number plate recognition (NPR) can be used in various fields such as vehicle tracking, traffic monitoring, automatic payment of tolls on highways or bridges, surveillance systems, tolls collection points, and parking management systems.

The escalating increase of contemporary urban and national road networks over the last decades emerged the need for efficient monitoring and management of road traffic. Meanwhile, rising vehicle use causes social problems such as accidents, traffic congestion, and consequent traffic pollution. Number Plate Recognition is a process where vehicles are identified or recognized using their number plate or license plate. NPR uses image processing techniques so as to extract the vehicle number plate from digital images.

- **Skills required :**

Python,Flask Integration,API

- **Project Objectives :**

1. What is an API?
2. How API can be integrated into web application
3. How to hit the APIs, send and retrieve the data
4. How to build web applications using flask framework

● Project Explanation:

To start off the project, firstly we need to set the following things up:

1. Installing Anaconda Navigator.
2. Installing required python packages
3. Rapid API account creation
4. Subscription of Application oriented API
5. Testing RapidAPI endpoint

After subscribing to the "License Plate Recognition" API , this shows up

The screenshot shows the RapidAPI website interface for the "License Plate Recognition" API. The page title is "License Plate Recognition API Documentation". Below the title, a brief description states: "License plate recognition involves capturing photographic video or images of license plates, whereby they are processed by a series of algorithms that are able to provide an alpha numeric conversion of the captured license plate images into a text entry."

The main content area is divided into two columns. The left column contains a search bar and a list of endpoints, with the selected endpoint being "POST /recognize_url". The right column displays the details for this endpoint, including a "Test Endpoint" button and a "Code Snippets" tab. The "Code Snippets" tab shows a Python request example using the 'requests' library.

Header Parameters:

Parameter	Value	Required
X-RapidAPI-Key	1684fa1d6dmsh60c658802264812p1b0bf3jsn58757764e534	REQUIRED
X-RapidAPI-Host	zyanyatech1-license-plate-recognition-v1.p.rapidapi.com	REQUIRED

Code Snippets (Python Requests):

```
import requests

url = "https://zyanyatech1-license-plate-recognition-v1.p.rapidapi.com/recognize_url"

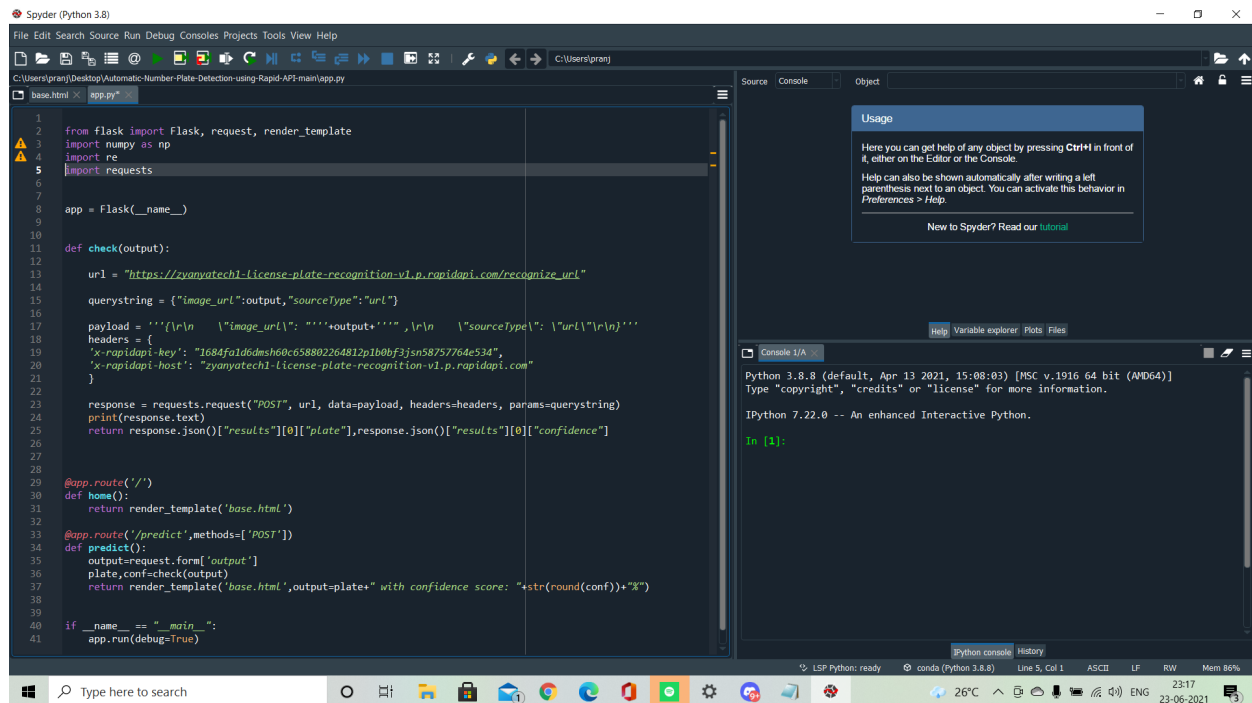
querystring = {"image_url": "http://eslamoda.com/wp-content/uploads/sites/2/2014/11/america-carro-600x600.jpg"}

headers = {
    'x-rapidapi-key': "1684fa1d6dmsh60c658802264812p1b0bf3jsn58757764e534",
    'x-rapidapi-host': "zyanyatech1-license-plate-recognition-v1.p.rapidapi.com"
}

response = requests.request("POST", url, headers=headers, params=querystring)

print(response.text)
```

Now from the above window, copy the code from "Code Snippets" and paste it in "Spyder" and then test it.



```
1 from flask import Flask, request, render_template
2 import numpy as np
3 import re
4 import requests
5
6
7
8 app = Flask(__name__)
9
10
11 def check(output):
12
13     url = "https://zyanyatech1-license-plate-recognition-v1.p.rapidapi.com/recognize_url"
14
15     querystring = {"image_url":output,"sourceType":"url"}
16
17     payload = '''{\n\n  \image_url\': \'''+output+'''\n\n  \sourceType\': \image_url\'''\n\n}'''
18     headers = {
19         'x-rapidapi-key': '1684fa1d6dms60c658802264812p1b0bf3jsn5875776de534',
20         'x-rapidapi-host': 'zyanyatech1-license-plate-recognition-v1.p.rapidapi.com'
21     }
22
23     response = requests.request("POST", url, data=payload, headers=headers, params=querystring)
24     print(response.text)
25     return response.json()["results"][0]["plate"],response.json()["results"][0]["confidence"]
26
27
28
29 @app.route('/')
30 def home():
31     return render_template('base.html')
32
33 @app.route('/predict',methods=['POST'])
34 def predict():
35     output=request.form['output']
36     plate,conf=check(output)
37     return render_template('base.html',output=plate+" with confidence score: "+str(round(conf)+"%")
38
39
40 if __name__ == "__main__":
41     app.run(debug=True)
```

In this code ,

Firstly, we have to import the libraries required for our project. (lines :2-5)

Next, we have to define the app function (line:8)

And next we have to define the function that calls our API via the web app.This function puts the input data provided by the user as the data to hit the API and get the desired results.(lines:11-25)

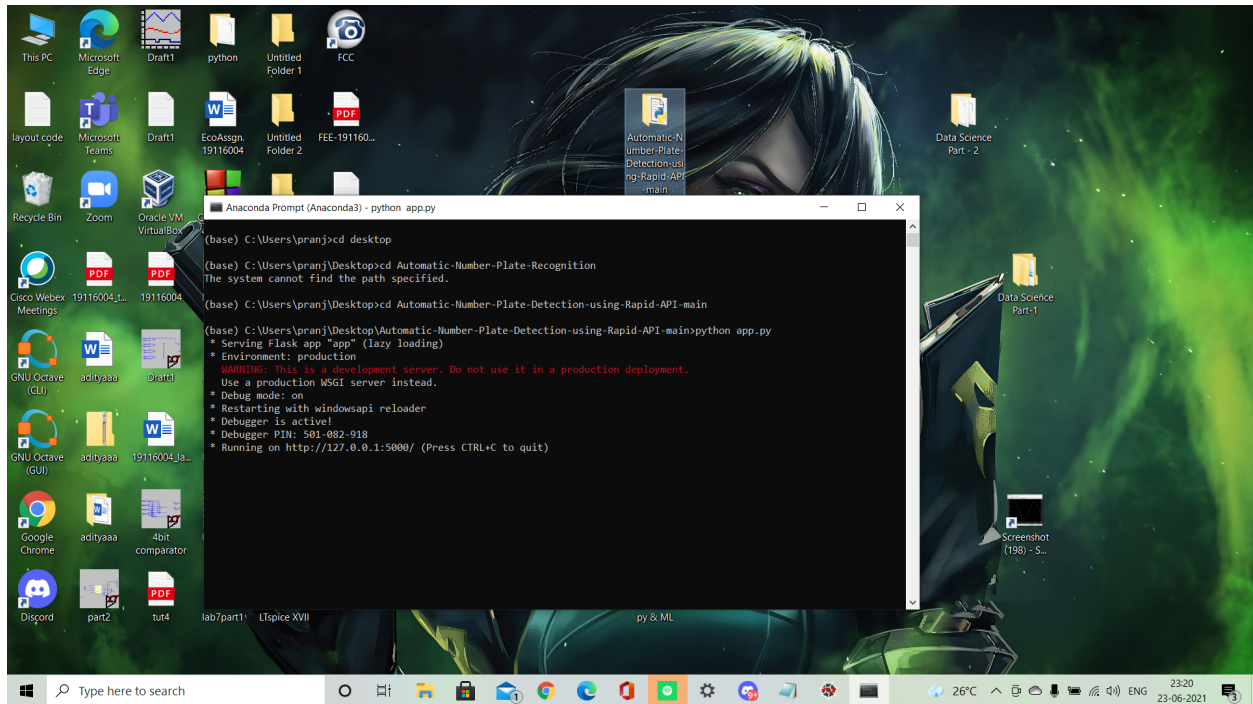
The next part is the routing of HTML page.(lines:29-31)

This part configures the results page by defining the 'predict()' function. This 'predict()' function extracts the numbers and alphabets from the Number-plate.(lines:33-37)

This part is 'calling of the main function'. Using the run method we can launch the flask application.(lines:40-41)

Now save this code as .py file and name it as 'app.py'.

Now run this code using Anaconda Prompt,



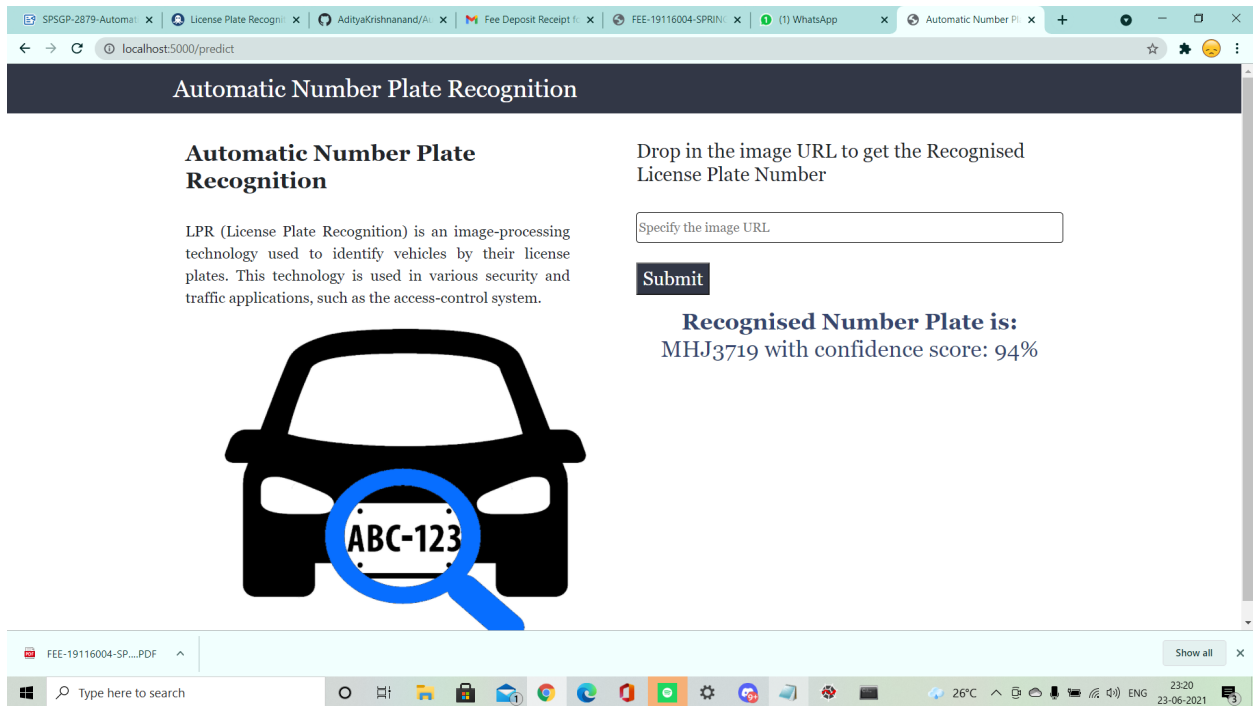
Now copy the URL and run it on chrome.

After running it on chrome , our UI will look like this along with the input provided and the output is produced with a confidence score of 94%

Here I have given this image as the input ,



The output UI will look like this :



Our output contains the number plate with confidence score. Confidence score may vary due to these factors :

- Broken number plate
- Blurry images
- Low resolution of the characters
- Similarity between certain characters like O and D , 5 and S , 8 and B , O and 0 etc...

Github repo link :

<https://github.com/AdityaKrishnanand/Automatic-Number-Plate-Recognition>