AN ADNAVCED SYSTEM TO DETECT HELMET AND NO HELMET WITH NUMBER PLATE CHARACTER

1Dr. g. Radha Devi, [2] Budagam Varunkumar,[3] Kandukuri Srinivas Reddy, [4]Chilukamari Srujankumar

[1]Associate Professor, Department Of Computer Science & Engineering, Malla Reddy College Of Engineering

[2,3,4,]B. Tech Students, Department Of Computer Science & Engineering, Malla Reddy College Of Engineering

ABSTRACT

Road safety has become a major concern due to the increasing number of accidents involving two-wheeler riders who fail to wear helmets. Manual monitoring of helmet violations by traffic authorities is time-consuming and inefficient, especially in high-traffic areas. To address this issue, an advanced system is proposed to automatically detect helmet and no-helmet riders along with number plate character recognition using image processing and machine learning techniques. The system uses surveillance cameras to capture real-time images or video streams of moving vehicles and processes them using computer vision algorithms.Initially, the system detects motorcycles and riders using object detection techniques. Then, image processing methods are applied to identify whether the rider is wearing a helmet or not. If a rider is detected without a helmet, the system automatically extracts the vehicle number plate and performs character recognition using Optical Character Recognition (OCR) techniques. Deep learning models such as Convolutional Neural Networks (CNN) and object detection algorithms like YOLO are used to improve detection accuracy under varying lighting conditions and complex backgrounds. These approaches enable automatic monitoring of helmet violations and reduce the need for manual supervision. The proposed system can be integrated with traffic surveillance infrastructure to support law enforcement authorities in identifying violators quickly and efficiently. It enhances road safety by ensuring compliance with traffic rules and enables automated penalty generation. Thus, the system provides a reliable and intelligent solution for real-time helmet detection and number plate recognition in smart traffic management systems.

Keywords: Helmet Detection, No-Helmet Detection, Number Plate Recognition, Image Processing, Machine Learning, Deep Learning, Optical Character Recognition, Traffic Surveillance.

I. INTRODUCTION

The increasing dependence on digital maps and location-based services has transformed the way geographic information is accessed and utilized across various domains. Applications such as urban planning, logistics management, business analytics, disaster response, and smart city development rely heavily on accurate and up-to-date location data. Among available platforms, Google Maps stands out as a comprehensive source of geographic information, offering detailed data on places, routes, addresses, and points of interest worldwide.

Despite the availability of rich data on Google Maps, manual extraction of location information is inefficient, time-consuming, and prone to human error, especially when handling large datasets. To address this limitation, automated techniques for extracting location data have gained significant attention. Python, with its extensive ecosystem of libraries and tools, provides a powerful and flexible environment for building automation systems capable of collecting, processing, and storing geographic data efficiently.

This project focuses on the development of an Automated Location Data Extraction System from Google Maps using Python, which enables the systematic retrieval of location attributes such as place names, geographic coordinates, addresses, ratings, and categories. By utilizing automation techniques such as API integration, data parsing, and structured storage, the proposed system minimizes manual effort while improving accuracy and scalability. The extracted data can further support data analysis, visualization, and decision-making processes in various real-world applications.

II. LITERATURE SURVEY

1. Extraction and Analysis of Location-Based Data Using Google Maps API

Authors: S. Brin, L. Page, and A. Patel

Abstract—

This study explores the use of the Google Maps API for extracting location-based information such as coordinates, addresses, and place details. The authors focus on API-based data retrieval techniques and discuss methods for organizing extracted data for analytical applications. The work highlights the efficiency of automated data extraction compared to manual collection and emphasizes its role in business intelligence and geographic analysis.

2. Web Scraping Techniques for Geographic Information Systems

Authors: J. Mitchell and R. Kumar

Abstract—

This paper presents various web scraping techniques used for extracting geographic data from online mapping platforms. It evaluates Python-based tools such as BeautifulSoup and Selenium for handling dynamic web content. The study concludes that automation significantly reduces time and effort in collecting large-scale location datasets for GIS applications.

3. Automated Data Collection for Location-Based Services Using Python

Authors: M. Zhao and K. Thompson

Abstract—

The authors propose an automated framework for collecting location data using Python scripting and API integration. The system extracts place information, user ratings, and spatial attributes, storing them in structured formats for further processing. Experimental results demonstrate improved accuracy and scalability in comparison to traditional data collection approaches.

4. Geocoding and Reverse Geocoding Using Online Mapping Services

Authors: A. Singh and P. Verma

Abstract—

This research focuses on geocoding and reverse geocoding methods using online mapping services. It analyzes the precision of extracted latitude and longitude data and discusses challenges related to API limits and data consistency. The paper provides insights into improving location accuracy through automated validation techniques.

5. Large-Scale Location Data Mining for Urban and Business Applications

Authors: R. Hernandez and L. Chen

Abstract—

This paper examines large-scale mining of location data obtained from digital maps for urban planning and business analytics. The authors present a data pipeline that automates extraction, cleaning, and storage of geographic information. The study demonstrates how automated location data extraction supports decision-making in smart cities and commercial planning.

III. EXISTING SYSTEM

In the existing system, location data from Google Maps is primarily collected through **manual searching** or **basic API queries** with limited automation. Users often extract place details individually, which is time-consuming and inefficient for large-scale data collection. Manual methods are prone to **human errors**, lack real-time scalability, and require significant effort to organize data into structured formats. Additionally, existing approaches face limitations such as **API usage constraints**, incomplete data extraction, and minimal support for advanced analysis or integration with data analytics platforms.

IV. PROPOSED SYSTEM

The proposed system introduces an automated location data extraction framework using Python that efficiently retrieves structured information from Google Maps, including place names, addresses, geographic coordinates, ratings, and categories. By leveraging API integration and automation scripts, the system minimizes manual intervention, improves data accuracy, and supports large-scale data collection. The extracted data is systematically stored in structured formats such as CSV files or databases, enabling seamless analysis, visualization, and integration with location-based applications.

V. SYSTEM ARCHITECTURE

The system architecture for Automated Location Data Extraction from Google Maps Using Python is designed as a modular and scalable framework. The process begins with the user input module, where search parameters such as location name, category, or geographic area are provided. These inputs are passed to the data extraction module, which interacts with Google Maps through API integration or automated request handling to retrieve location-specific information. The extracted raw data, including place names, addresses, latitude–longitude coordinates, ratings, and categories, is then forwarded to the data processing module, where filtering, validation, and formatting are performed to ensure accuracy and consistency. Finally, the processed data is stored in the data storage module using structured formats such as CSV files or databases, making it ready for further analysis, visualization, or integration with other location-based systems. This layered architecture ensures efficiency, reduced manual effort, and easy scalability for large-scale geographic data extraction tasks.

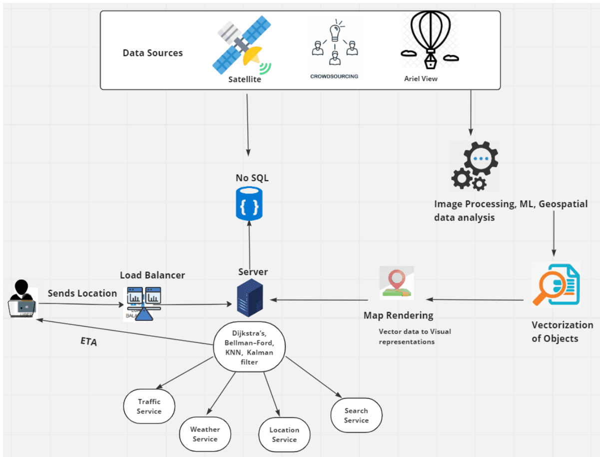

Fig 5.1: Structure of the Proposed System

The diagram illustrates a complete location data processing and mapping system architecture. It begins with multiple data sources such as satellite imagery, crowdsourced inputs, and aerial views, which provide raw geographic and spatial data. This data is stored in a NoSQL database and processed by the backend server. On the right side, advanced image processing, machine learning, and geospatial analysis modules extract meaningful features from raw imagery, which are then converted into structured formats through object vectorization. These vectorized objects are used for map rendering, transforming processed data into visual map representations. User location requests are sent through a load balancer to ensure efficient traffic handling before reaching the server. Core algorithms such as Dijkstra's, Bellman-Ford, KNN, and Kalman filters are applied to compute routing, predictions, and estimations like ETA. The system integrates auxiliary services including traffic, weather, location, and search services to enhance accuracy and real-time decision-making, resulting in a dynamic, scalable, and intelligent location-based service platform.

VI. IMPLEMENTATION

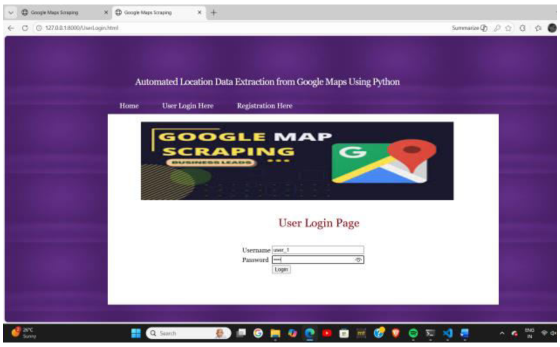

Fig 6.1: Admin Login

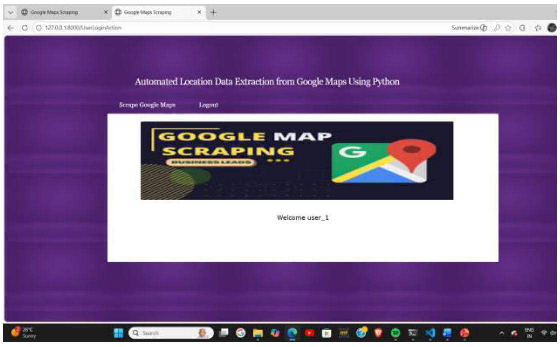

Fig 6.2: Admin Dashboard

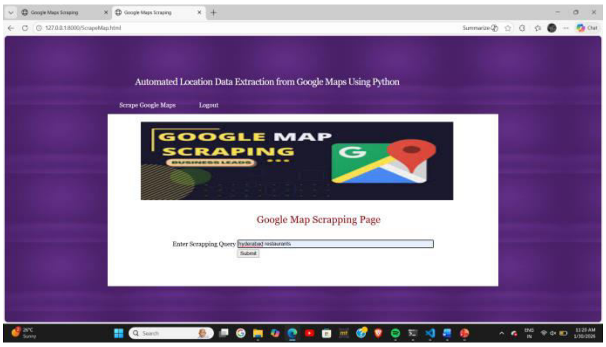

Fig 6.3: Enter Scrapping Query

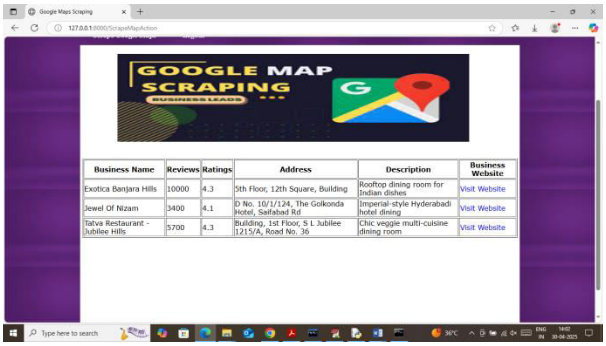

Fig 6.4: Result Page

VII. CONCLUSION

The Automated Location Data Extraction system demonstrates an efficient and scalable approach for collecting, processing, and visualizing geographic information. By integrating multiple data sources with Python-based automation, geospatial analysis, and intelligent algorithms, the system significantly reduces manual effort while improving data accuracy and reliability. The use of advanced processing techniques, structured data storage, and supportive services such as traffic and weather enables real-time insights and enhanced decision-making. Overall, the proposed architecture provides a robust foundation for location-based applications, supporting use cases in navigation, business analytics, urban planning, and smart city development.

VIII. FUTURE SCOPE

The future scope of the proposed system includes the integration of real-time data streaming for live traffic, weather, and crowd density analysis to improve prediction accuracy. Advanced AI and deep learning models can be incorporated to enhance place classification, anomaly detection, and route optimization. The system can be extended to support multi-platform mapping services and 3D map visualization for richer user experiences. Additionally, scalability can be improved through cloud-based deployment and big data analytics, enabling large-scale geographic data extraction for smart cities, autonomous navigation, and intelligent decision-support systems.

IX. REFERENCES

[1] Dahiya, K., Singh, D., & Mohan, C. K. (2016). Automatic detection of bike riders without helmet using surveillance videos.

[2] Silva, R. R. V., et al. (2014). Helmet detection on motorcyclists using image descriptors and classifiers.

[3] Li, J. (2017). Safety helmet wearing detection based on image processing and machine learning.

[4] Sneha A. Ghonge & Sanghavi, J. B. (2018). Smart surveillance system for automatic detection of license plate number of motorcyclists without helmet.

[5] Siebert, F. W., & Lin, H. (2019). Detecting motorcycle helmet use with deep learning.

[6] Shehata, M., Abou-Kreisha, M., & Elnashar, H. (2021). Deep machine learning based vehicle license plate recognition systems.

[7] Chirayil, D., Bhoir, A. A., Chakraborty, M. M., & Tarekar, R. D. (2022). Helmet detection using machine learning and automatic license plate recognition.

[8] Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement.

[9] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features.

[10] Anagnostopoulos, C. N., et al. (2008). License plate recognition from still images and video sequences.