VISVESVARAYATECHNOLOGICAL UNIVERSITY
JNANASANGAMA,BELAGAVI-590018

IV SEMESTER ACTIVITYREPORT 2022-23

ON

"Application of Mathematics in the field of Machine Learning."

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

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by

AKSHAY VIVEKANANDA B-1KS21CS011

GAANA S-1KS21CS029

HARSHITHA P – 1KS21CS034

Course: **Mathematical Foundations for Computing, Probability & Statistics**

Course Code: **21MATCS41**

Course- In-Charge: **Mr. Naveen V**

DepartmentofComputerScience&Engineering
K.S.INSTITUTEOFTECHNOLOGY
No.14,Raghuvanahalli,KanakapuraRoad,Bengaluru-
560109

K.S. INSTITUTE OF TECHNOLOGY

BENGALURU-560109

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the Activity entitled **"Application of Mathematics in the field of Machine Learning"** presented by **Akshay Vivekananda B, Gaana S & Harshitha P**, **USN: 1KS21CS011, 1KS21CS029, 1KS21CS034** of **IV** semester in partial fulfilment of the award of Bachelor of Engineering in CSE in Visvesvaraya Technological University, Belagavi during the academic year **2023**. The Activity has been approved as it satisfies the academic requirements in respect of Activity work prescribed for the Bachelor of Engineering degree.

2/9/23

Mr. Naveen V
Professor
Department of Mathematics
K.S.I.T., Bengaluru

2/9/23

Dr. Rekha B.Venkatapur
Head of Department,
CSE
K.S.I.T., Bengaluru

Dr. Dilip Kumar.K
Principal / Director,
K.S.I.T., Bengaluru

DECLARATION

I, **Akshay Vivekananda B, Gaana S & Harshitha P** bearing the **USN: 1KS21CS011, 1KS21CS029, 1KS21CS034** studying in the 4th semester of Bachelor of Engineering in Computer Science and Engineering at K S Institute of Technology, Bengaluru, hereby declare that the Activity Report entitled **"Application of Mathematics in the field of Machine Learning"**, is a record of original work based on Course Activity. Under the Internal Guidance of **Mr. Naveen V**, Assistant Professor, Department of Mathematics, K. S. Institute of Technology. The Activity Report has been submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science and Engineering. The results embodied in this report have not been submitted to any other University or Institute for the award of any degree.

Place: Bengaluru

Date:

NAME (USN)

AKSHAY VIVEKANANDA B (1KS21CS011)

GAANA S (1KS21CS029)

HARSHITHA P (1KS21CS034)

SUMMARY

The project on live data tracking in Python presents a comprehensive methodology for building real-time data tracking systems. It encompasses the entire data lifecycle, from the identification of data sources to secure visualization. Python's versatility and extensive library support are key elements in this endeavor, enabling the collection, processing, and visualization of data from various sources. The methodology emphasizes data quality, security, and scalability, offering a robust foundation for applications across industries such as IoT, logistics, healthcare, and finance.

This project acknowledges the critical role of Python in the realm of real-time data tracking, where it serves as a versatile and accessible tool. By implementing this methodology, organizations and individuals can effectively monitor assets, analyze trends, and make data-driven decisions. It fosters proactive responses to changing conditions, ultimately enhancing productivity and competitiveness. With Python's continual advancements and the evolving landscape of data analytics, this project empowers users to stay at the forefront of real-time data tracking, ensuring adaptability to emerging technologies and harnessing the transformative potential of live data in today's dynamic world.

ACKNOWLEDGEMENT

The successful completion of the seminar would be incomplete without the mention of the people who made it possible and whose constant guidance crowned our effort with success.

We take this opportunity to express our sincere gratitude to our Management K S Institute of Technology, Bengaluru for providing the environment to present the seminar.

We would like to extend our gratitude to **Dr. Dilip Kumar** Principal & Director, K.S. Institute of Technology, Bengaluru, for providing an opportunity to present the seminar.

We thank **Dr. Rekha. B. Venkatapur**, Professor and Head, Department of Computer Scienceand Engineering, K.S. Institute of Technology, Bengaluru, for her encouragement.

We thank Course Coordinator, **Mr. Naveen V**, Assistant Professor Department of Basic Science, K.S. Institute of Technology, Bengaluru, for his constant support and guidance.

Finally, we would like to thank all the teaching and non-teaching staff of the college for their co-operation. Moreover, we thank all our families and friends for their invaluable support and cooperation.

ABSTRACT

The project on live data tracking in Python represents a dynamic and vital approach to harnessing the power of real-time data in an increasingly data-centric world. It introduces a comprehensive methodology that guides the development of systems capable of continuously collecting, processing, and visualizing data from various sources. With Python as its foundation, the project offers flexibility, versatility, and accessibility, making it an invaluable resource for organizations and individuals across multiple domains, including IoT, logistics, healthcare, finance, and more.

This project recognizes Python as a driving force in real-time data tracking, enabling users to create robust systems that emphasize data quality, security, and scalability. By following this methodology, stakeholders can not only monitor assets and analyze trends but also make data-driven decisions with agility and precision. In an era where timely information is paramount, this project empowers users to remain at the forefront of data analytics, adapt to emerging technologies, and leverage the transformative potential of live data to drive innovation and enhance competitiveness.

TABLE OF CONTENTS

INTRODUCTION TO MACHINE LEARNING

Machine learning (ML) is an umbrella term for solving problems for which the development of algorithms by human programmers would be cost-prohibitive, and instead the problems are solved by helping machines 'discover' their 'own' algorithms, without needing to be explicitly told what to do by any human-developed algorithms. Recently, generative artificial neural networks have been able to surpass the results of many previous approaches. Machine learning approaches have been applied to large language models, computer vision, speech recognition, email filtering, agriculture, and medicine, where it is too costly to develop algorithms to perform the needed tasks. The mathematical foundations of ML are provided by mathematical optimization (mathematical programming) methods. Data mining is a related (parallel) field of study, focusing on exploratory data analysis through unsupervised learning.ML is known in its application across business problems under the name of predictive analytics. Although not all machine learning is statistically based, computational statistics is an important source of the field's methods.

Machine learning is a subfield of artificial intelligence (AI) that focuses on the development of algorithms and models that enable computers to learn and improve from experience without being explicitly programmed. The core idea behind machine learning is to develop mathematical models and algorithms that allow computers to learn from examples or data without being explicitly programmed for specific tasks.

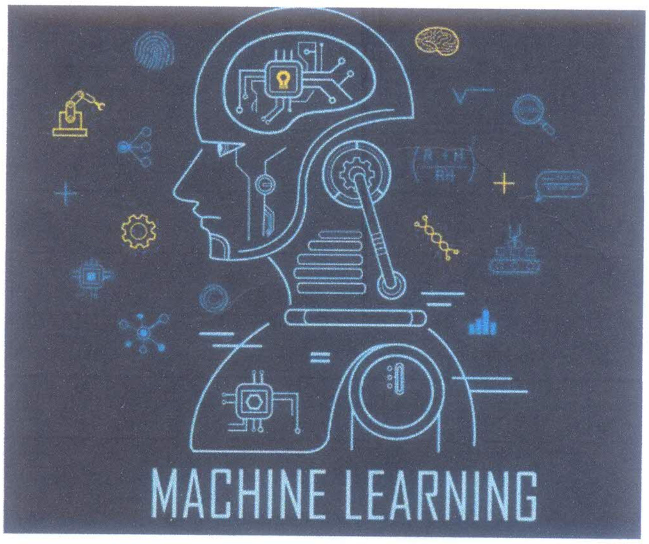


FIGURE 1 : MACHINE LEARNING

ROLE OF MATHEMATICS IN MACHINE LEARNING

Mathematics plays a fundamental and indispensable role in machine learning (ML). ML is essentially a mathematical discipline that relies on various mathematical concepts, techniques, and algorithms to enable computers to learn from data and make predictions. Here's an exploration of the key ways mathematics is involved in ML:

1. Linear Algebra:

- Role: Linear algebra is essential for understanding and working with high-dimensional data, which is common in ML. It provides the foundation for various operations on vectors and matrices.
- Applications: Matrix multiplication, eigende composition, singular value decomposition (SVD), and solving systems of linear equations are used in techniques like Principal Component Analysis (PCA) and Singular Value Decomposition.

2. Calculus:

- Role: Calculus is used to optimize ML models. It helps in finding the minimum or maximum of functions, which is crucial for training models and minimizing prediction errors.
- Applications: Gradient descent, a key optimization algorithm in ML, relies on calculus to adjust model parameters and minimize a loss function.

3. Probability and Statistics:

- Role: Probability and statistics are the bedrock of statistical learning theory. They help quantify uncertainty, estimate parameters, and assess model performance.
- Applications: Probability distributions are used to model uncertainty, and statistical tests are employed to assess the significance of findings. Bayesian statistics is also used for probabilistic modeling.

4. Information Theory:

- Role: Information theory concepts, such as entropy and mutual information, are used in feature selection, dimensionality reduction, and designing loss functions.

- Applications: For example, decision trees use information gain to determine the most informative features for splitting data.

5. Optimization:

- Role: Optimization techniques are crucial for finding the best model parameters that minimize a cost or loss function.
- Applications: Algorithms like gradient descent and its variants are used to iteratively optimize model parameters during the training process.

6. Linear Regression and Logistic Regression:

- Role: These are foundational ML algorithms that are based on mathematical concepts. Linear regression models relationships between variables, while logistic regression models binary outcomes.
- Applications: Linear and logistic regression are widely used in various fields, including finance, healthcare, and social sciences.

7. Matrix Factorization:

- Role: Matrix factorization methods decompose large matrices into smaller, more interpretable matrices, helping to uncover hidden patterns in data.
- Applications: Collaborative filtering for recommendation systems and dimensionality reduction methods like PCA are based on matrix factorization.

8. Numerical Analysis:

- Role: Numerical analysis ensures the stability and efficiency of ML algorithms, especially when dealing with large datasets or complex models.
- Applications: Solving systems of equations, approximating derivatives, and ensuring numerical stability are crucial for implementing ML algorithms.

Mathematics is the language of machine learning, providing the theoretical foundation and practical tools for understanding, developing, and optimizing ML algorithms. A strong mathematical background is essential for data scientists and ML engineers to effectively work with machine learning models and make informed decisions in the field.

METHODOLOGY

The methodology for live data tracking involves a systematic approach to continuously collect, process, and visualize real-time data. Here's a paragraph outlining the key steps:

To implement live data tracking, a well-defined methodology is essential. It begins with the deployment of data collection sources, such as sensors, IoT devices, or data streams. These sources generate continuous data, which is then transmitted to a central repository or server. In the server, the data is processed and stored in a structured format, often using databases or data warehouses. Real-time processing techniques, like stream processing or complex event processing, may be employed to analyze and filter incoming data for relevance and accuracy. Once processed, the data can be visualized through web-based dashboards or reporting tools, enabling users to monitor and make informed decisions based on the live data.

Continuous monitoring of the system's performance and data quality is integral, with mechanisms in place for data validation, error handling, and redundancy to ensure the reliability and accuracy of the live data tracking system. This methodology serves as a foundation for applications across various domains, including IoT, logistics, healthcare, and financial markets, where timely and accurate data insights are critical.

Furthermore, data security and privacy measures are of utmost importance, particularly when dealing with sensitive information. Implementing encryption, access controls, and compliance with data protection regulations are essential components to safeguard data integrity and user privacy.

Continual improvement and maintenance are vital aspects of the live data tracking methodology. Regularly updating data collection sources, optimizing data processing pipelines, and enhancing data visualization tools are necessary to ensure the system's effectiveness. Scalability considerations must also be taken into account to accommodate the growth of data sources and increasing data volumes.

The methodology for live data tracking encompasses the entire data lifecycle, from collection and processing to visualization and maintenance. A well-structured and adaptable approach is key to harnessing the benefits of real-time data insights across various domains, contributing to informed decision-making and improved operational efficiency.

PYCHARM

FIGURE 2 : PYCHARM

One of PyCharm's key advantages is its ability to improve code quality. The IDE encourages best coding practices through features like code analysis and automated refactoring tools, which contribute to cleaner and more maintainable codebases. Developers also appreciate its versatility, as PyCharm is not limited to Python alone. It offers robust support for web development technologies such as HTML, CSS, and JavaScript, making it a valuable asset for full-stack developers.

PyCharm's availability in both Community and Professional editions caters to developers with varying needs and budgets. This accessibility ensures that both beginners and experienced programmers can benefit from its rich feature set. JetBrains, the company behind PyCharm, actively maintains and updates the IDE, ensuring compatibility with the latest Python versions and providing timely support for any issues that may arise.

The PyCharm community is vibrant and supportive, with an extensive ecosystem of plugins and extensions that enable developers to tailor the IDE to their specific requirements. This adaptability, combined with its educational tools designed for teaching and learning Python, further cements PyCharm's reputation as an indispensable tool in the Python development landscape.

PyCharm has earned its place as a powerful and reliable integrated development environment for Python. Its comprehensive feature set, including a smart code editor, debugger, VCS integration, and support for web development technologies, significantly streamlines the development process. Whether you are a novice or an experienced Python programmer, PyCharm is a versatile and dependable companion that facilitates efficient, error-free, and high-quality Python coding.

LIBRARIES IN PYCHARM

PyCharm is a popular integrated development environment (IDE) for Python, and it supports a wide range of libraries and frameworks that enhance your Python development experience. Here are some of the most commonly used libraries and frameworks that you can work with in PyCharm:

1. NumPy: NumPy is a fundamental library for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a variety of high-level mathematical functions to operate on these arrays.

2. Pandas: Pandas is a data manipulation and analysis library. It offers data structures like DataFrames and Series, making it easy to handle and analyze structured data.

3. Matplotlib: Matplotlib is a popular data visualization library. It allows you to create a wide range of static, animated, and interactive plots and charts.

4. Seaborn: Seaborn is a data visualization library based on Matplotlib. It provides a high-level interface for creating attractive and informative statistical graphics.

5. scikit-learn: scikit-learn is a machine learning library for Python. It includes a wide range of machine learning algorithms and tools for tasks like classification, regression, clustering, and more.

6. TensorFlow and PyTorch: These libraries are essential for deep learning and neural network development. TensorFlow is developed by Google, while PyTorch is supported by Facebook's AI Research lab. Both are commonly used for building and training neural networks.

7. Django and Flask: Django and Flask are popular web frameworks for Python. They enable you to build web applications and RESTful APIs. PyCharm provides excellent support for web development with these frameworks.

8. Requests: The Requests library is commonly used for making HTTP requests and handling HTTP responses. It simplifies working with web APIs.

9. Beautiful Soup: Beautiful Soup is a library for web scraping and parsing HTML and XML documents. It's useful for extracting data from web pages.

10. SQLAlchemy: SQLAlchemy is an SQL toolkit and Object-Relational Mapping (ORM) library. It

facilitates working with databases and supports multiple database systems.

11. pytest: pytest is a popular testing framework for Python. It simplifies the process of writing and running tests for your Python code.

12. Flask-RESTful: If you're building RESTful APIs with Flask, Flask-RESTful is an extension that simplifies the process of creating API endpoints and handling requests.

13. Jupyter Notebook: While not a library, Jupyter Notebook is a widely used tool for interactive computing and data exploration. PyCharm provides Jupyter Notebook integration.

To use these libraries in PyCharm, you typically need to install them using the Python package manager, pip, and then import them into your Python scripts or projects. PyCharm's package manager integration and code completion features make it easy to work with these libraries efficiently.

SUPPORT VECTOR MACHINES

Support vector machine (SVM) is a machine learning algorithm that uses supervised learning models to solve complex classification, regression, and outlier detection problems by performing optimal data transformations that determine boundaries between data points based on predefined classes, labels, or outputs. SVMs can handle linearly separable data by using a linear kernel, which defines the hyperplane as a linear combination of the input features. However, SVMs are also capable of handling non-linearly separable data by using a kernel function.

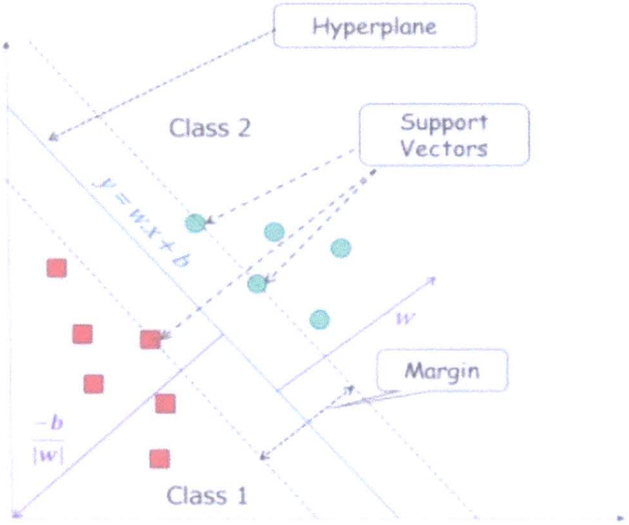
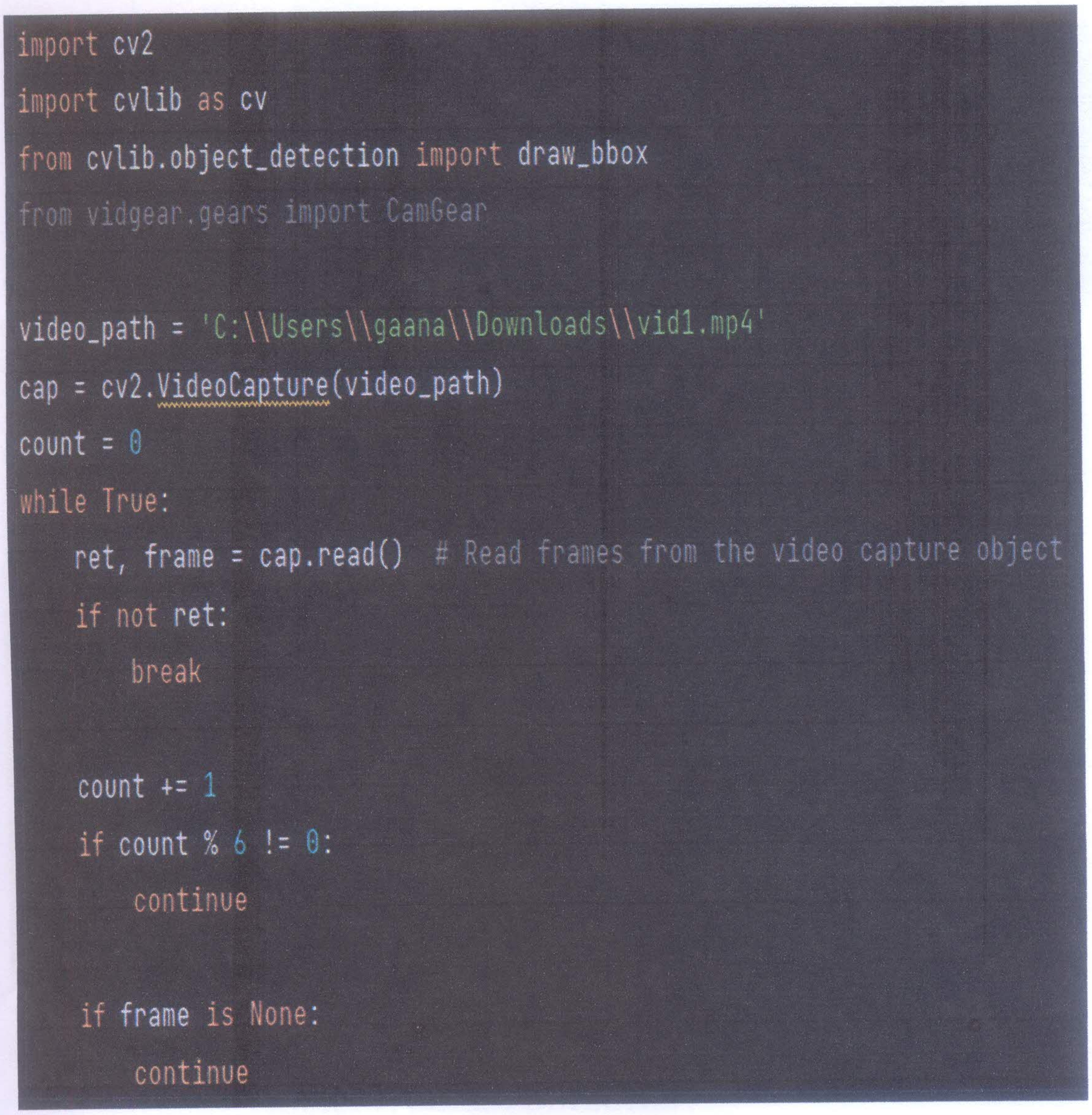


FIGURE 3: SUPPORT VECTOR MACHINES

OBJECT DETECTION

Object detection in Python is a computer vision task that involves identifying and locating objects within images or videos. It's a fundamental component of various applications, such as self-driving cars, security systems, and image-based search engines. You can perform object detection in Python using several libraries and frameworks. Here, we'll focus on using the popular library OpenCV and the deep learning framework TensorFlow for object detection.

CODE :

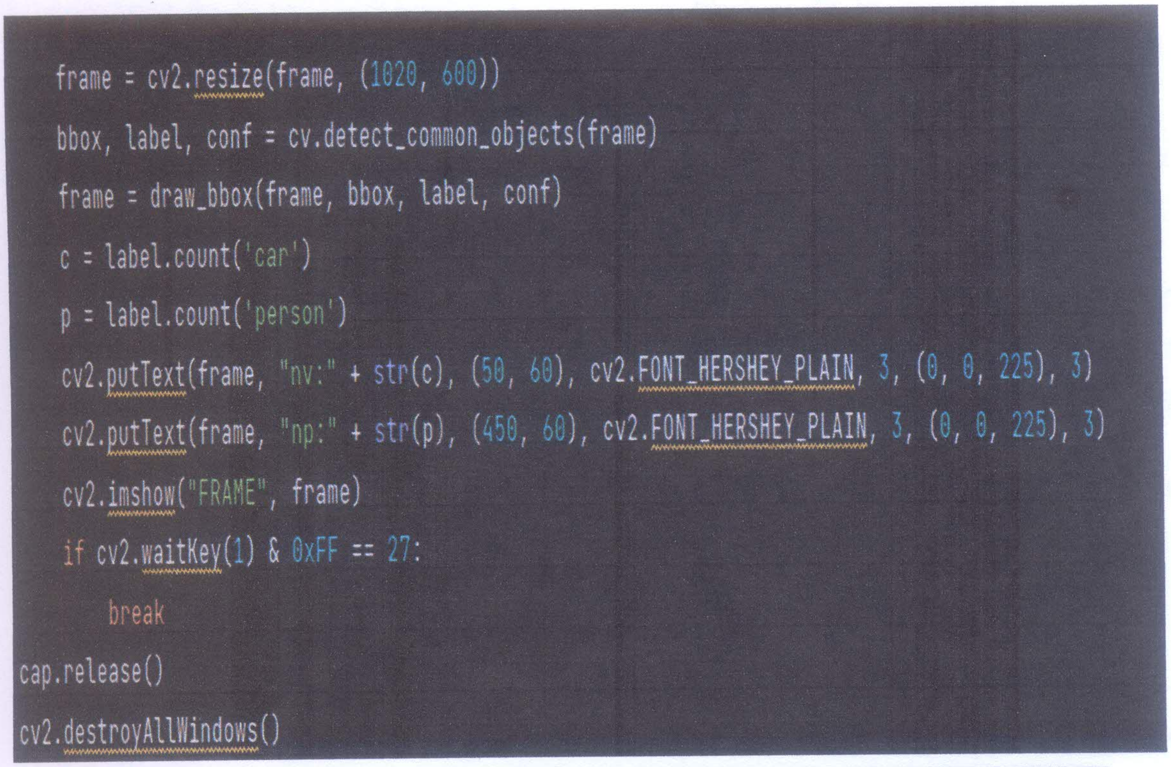


```
import cv2
import cvlib as cv
from cvlib.object_detection import draw_bbox
from vidgear.gears import CamGear

video_path = 'C:\\Users\\gaana\\Downloads\\vid1.mp4'
cap = cv2.VideoCapture(video_path)
count = 0
while True:
    ret, frame = cap.read()  # Read frames from the video capture object
    if not ret:
        break

    count += 1
    if count % 6 != 0:
        continue

    if frame is None:
        continue
```



```
    frame = cv2.resize(frame, (1020, 600))
    bbox, label, conf = cv.detect_common_objects(frame)
    frame = draw_bbox(frame, bbox, label, conf)
    c = label.count('car')
    p = label.count('person')
    cv2.putText(frame, "nv:" + str(c), (50, 60), cv2.FONT_HERSHEY_PLAIN, 3, (0, 0, 225), 3)
    cv2.putText(frame, "np:" + str(p), (450, 60), cv2.FONT_HERSHEY_PLAIN, 3, (0, 0, 225), 3)
    cv2.imshow("FRAME", frame)
    if cv2.waitKey(1) & 0xFF == 27:
        break
cap.release()
cv2.destroyAllWindows()
```

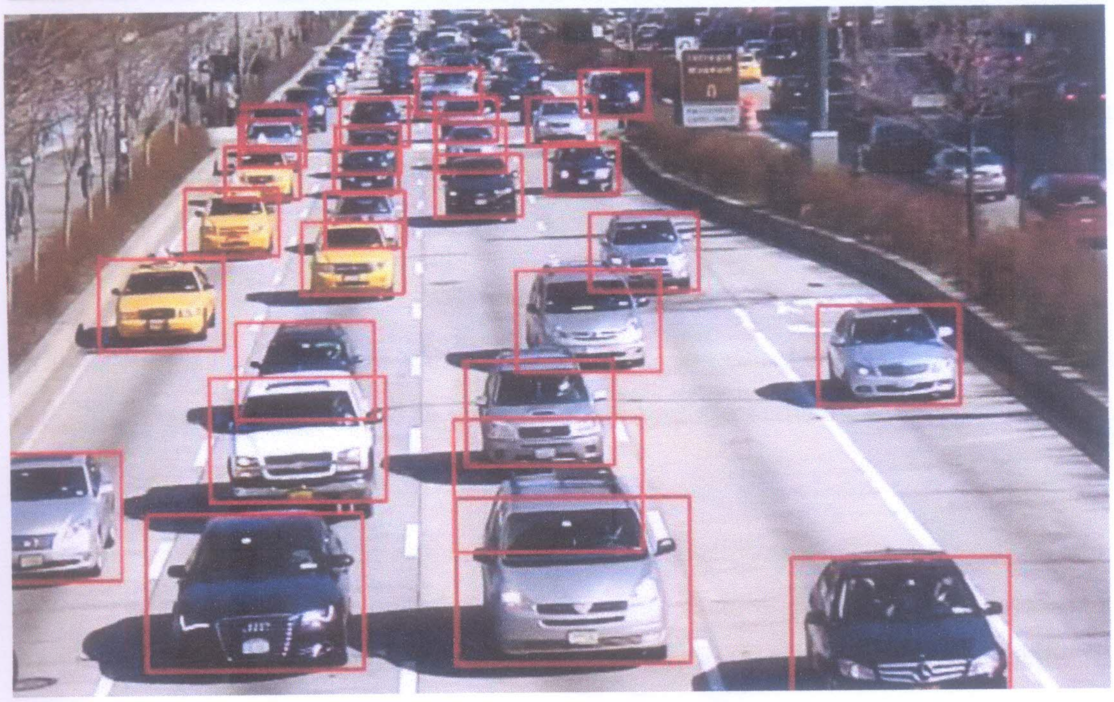


FIGURE 4 : OBJECT DETECTION

LIVE DATA TRACKING

Live data tracking in Python is a dynamic process that involves the continuous collection, processing, and visualization of real-time data from various sources. The methodology begins by identifying and establishing connections with data sources, which can range from IoT devices, sensors, APIs, to streaming data feeds. Python's versatility and rich ecosystem of libraries make it well-suited for handling diverse data sources.

Once the data sources are in place, Python scripts or applications come into play, responsible for collecting data in real-time. Libraries like `requests` or `paho-mqtt` facilitate data retrieval from web services or MQTT-based systems, while custom scripts can be employed to interface with sensor data.

Data processing is a pivotal step, where Python is used to clean, transform, and filter incoming data to ensure its accuracy and relevance. Depending on the nature of the data, real-time analytics and calculations can be performed using libraries like `pandas` and `numpy`.

To store the live data, databases such as PostgreSQL or specialized time-series databases like InfluxDB can be employed. Real-time processing tools like Apache Kafka or Flink can be integrated into the pipeline to handle high-volume data streams effectively.

For data visualization, Python offers an array of libraries including Matplotlib, Seaborn, and Plotly, enabling the creation of live and interactive visualizations. Web-based dashboards, powered by web frameworks like Flask or Django, allow users to access and interact with real-time data, providing a user-friendly interface.

Real-time alerts are crucial for immediate responses to critical events or predefined conditions in the data. Python scripts can be used to implement alerting mechanisms, sending notifications via email or SMS using libraries like smtplib or Twilio.

Security measures, including authentication, authorization, and data encryption, are paramount to protect the integrity and confidentiality of live data. Regular monitoring, logging, and testing ensure the system's performance, reliability, and data accuracy. Comprehensive documentation and ongoing maintenance round out the methodology, ensuring the live data tracking system remains robust and adaptable to evolving data sources and requirements.

CODE :

```
import cv2
import cvlib as cv
import traceback
from cvlib.object_detection import draw_bbox
from vidgear.gears import CamGear

try:
   stream = CamGear(source='https://www.youtube.com/watch?v=En_3pkxIJRM',
stream_mode=True,
            logging=True).start()  # YouTube Video URL as input
except:
   # printing stack trace
   traceback.print_exc()

count = 0
while True:
   frame = stream.read()
   count += 1
   if count % 6 != 0:
      continue

   frame = cv2.resize(frame, (1020, 600))
   bbox,label,conf=cv.detect_common_objects(frame)
   frame=draw_bbox(frame,bbox,label,conf)
   c=label.count('car')
   p=label.count('person')
   cv2.putText(frame,"nc :"+str(c),(50,85),cv2.FONT_HERSHEY_PLAIN,3,( 225,225,225),3)
   cv2.putText(frame,"np :"+str(p), (450,85), cv2.FONT_HERSHEY_PLAIN, 3, (225, 225, 225), 3)
   cv2.imshow("FRAME", frame)
   if cv2.waitKey(1) & 0xFF == 27:
      break
stream.release()
cv2.destroyAllWindows()
```

CONCLUSION

In conclusion, live data tracking in Python is a versatile and powerful endeavor that empowers organizations and individuals to harness the real-time information flow from diverse sources for informed decision-making and improved operational efficiency. Python's rich ecosystem of libraries, coupled with its flexibility and ease of use, make it an ideal choice for building and deploying real-time tracking systems. The methodology involves a holistic approach, from data source identification to secure data visualization, with a keen focus on data quality, security, and scalability.

As businesses increasingly rely on data-driven insights, live data tracking in Python serves as a critical tool across various domains, including IoT, logistics, healthcare, finance, and beyond. It enables organizations to monitor assets, analyze trends, and respond promptly to changing conditions, ultimately contributing to enhanced productivity and competitiveness. The continuous evolution of Python libraries and tools ensures that live data tracking remains at the forefront of modern data analytics, allowing users to adapt to emerging technologies and harness the power of real-time data in an ever-changing world.

REFERENCES

1. https://thepythoncode.com/article/real-time-object-tracking-with-yolov8-opencv.

2. https://stackoverflow.com/questions/75225743/live-location-tracking-using-python.

3. https://www.geeksforgeeks.org/opencv-python-program-vehicle-detection-video-frame/.

4. https://www.youtube.com/watch?v=En_3pkxIJRM%27