Chapter 32: Python Programs Using Data Handling and Visualization Libraries

Introduction

In this chapter, students will learn how to write simple Python programs that perform basic data processing and visualization tasks. Using Python libraries like `NumPy`, `Pandas`, `Matplotlib`, and `OpenCV`, we can analyze data, create visualizations such as line and scatter plots, and even work with image files. These skills form the basis for understanding AI and machine learning pipelines, which rely heavily on data manipulation and visualization.

32.1 Add Elements of Two Lists

Program Objective:

Write a Python program to add the corresponding elements of two lists.

Code:

```
list1 = [10, 20, 30, 40, 50]
list2 = [5, 15, 25, 35, 45]

# Adding elements
result = [a + b for a, b in zip(list1, list2)]

print("List 1:", list1)
print("List 2:", list2)
print("Sum of lists:", result)
```

32.2 Calculate Mean, Median and Mode Using NumPy

Program Objective:

Calculate mean, median, and mode using NumPy and SciPy libraries.

Code:

```
import numpy as np
from scipy import stats

data = [10, 20, 20, 30, 40, 50, 50, 50, 60]

mean = np.mean(data)
median = np.median(data)
```

```
mode = stats.mode(data)

print("Data:", data)
print("Mean:", mean)
print("Median:", median)
print("Mode:", mode.mode[0])
```

> ⚠ `scipy.stats.mode` returns a ModeResult object. We use `.mode[0]` to access the actual mode value.

32.3 Display a Line Chart from (2,5) to (9,10)

Program Objective:

Display a line chart using `Matplotlib`.

Code:

```
import matplotlib.pyplot as plt

x = [2, 3, 4, 5, 6, 7, 8, 9]
y = [5, 6, 7, 8, 9, 9.5, 10, 10]

plt.plot(x, y, marker='o', linestyle='-', color='blue')
plt.title("Line Chart")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.grid(True)
plt.show()
```

32.4 Display a Scatter Chart for Given Points

Program Objective:

Create a scatter plot of given data points.

Code:

```
import matplotlib.pyplot as plt

x = [2, 9, 8, 5, 6]
y = [5, 10, 3, 7, 18]

plt.scatter(x, y, color='red')
plt.title("Scatter Plot")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.grid(True)
plt.show()
```

32.5 Read a CSV File and Display 10 Rows

Program Objective:

Read and display the first 10 rows of a CSV file using `Pandas`.

Code:

```
import pandas as pd

# Replace 'filename.csv' with the actual path of your CSV file
df = pd.read_csv("filename.csv")
print(df.head(10))
```

> ⚠ Make sure `filename.csv` is saved in the same directory or provide the full file path.

32.6 Read a CSV File and Display Its Information

Program Objective:

Read a CSV file and display information such as column names, data types, and non-null values.

Code:

```
import pandas as pd

df = pd.read_csv("filename.csv")
print("Basic Information of the Dataset:\n")
print(df.info())
```

32.7 Read and Display an Image Using Python

Program Objective:

Read and display an image using `OpenCV`.

Code:

```
import cv2

# Replace 'image.jpg' with the actual image filename
img = cv2.imread('image.jpg')
```

```
# Display the image
cv2.imshow('Displayed Image', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

> ⚠ If you're using a Jupyter notebook, use `cv2.imshow()` with caution. Alternatively, display using matplotlib:

```
import matplotlib.pyplot as plt

plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.axis('off')
plt.show()
```

32.8 Identify the Shape of an Image Using Python

Program Objective:

Read an image and identify its dimensions (height, width, channels).

Code:

```
import cv2

img = cv2.imread('image.jpg')
print("Image shape (Height, Width, Channels):", img.shape)
```

> `img.shape` returns a tuple like `(height, width, channels)`. For grayscale images, it returns only height and width.

Summary

In this chapter, we practiced various Python programs that help in data handling and visualization—core concepts in Artificial Intelligence. Students learned to:

- Perform list operations.
- Calculate statistics like mean, median, and mode.
- Plot data using line and scatter charts.
- Read and explore CSV files using `Pandas`.
- Handle and analyze images using `OpenCV`.

These foundational skills are essential for any AI task that involves understanding and visualizing data.
