DAL DA5401 Assignment 1: Report

Utkarsh Pathak : DA24S011

11th August

1 Introduction

This report provides a detailed report on the operations performed on the matrix derived from a custom extarcted dataset. The operations include normalization, discretization, matrix creation, rotation, flipping, and visualization. The code processes data to produce scatter plots and matrix visualizations, demonstrating how matrix transformations affect data representation.

2 Data Loading and Preprocessing

2.1 Data Creation

The data is created by uploading the captured image to the website: `https://wpd.starrydata2.org/`. WebPlotDigitizer is a computer vision assisted software that helps extract numerical data from images of a variety of data visualizations. The data points can be extracted into a CSV file. See Figure 1.

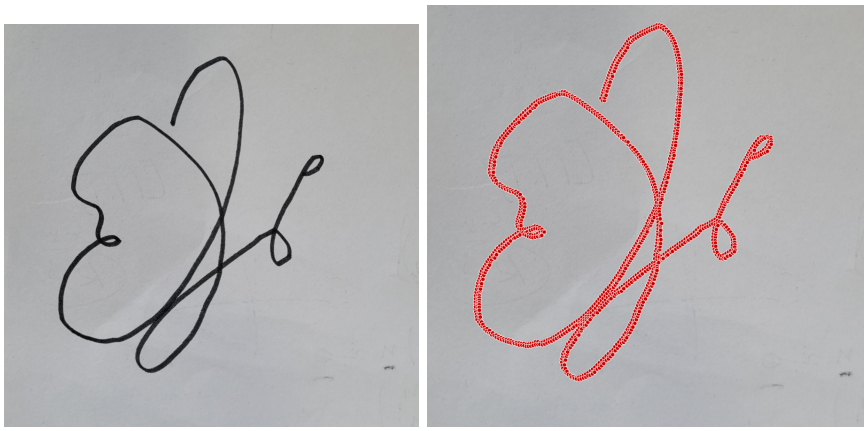


Figure 1: Digitised the original image using the online tool.

2.2 Data Loading

The data is loaded from a CSV file named `data.csv` using the Pandas library. The dataset is contains two columns representing x and y coordinates. They are renamed to X and Y.

2.3 Normalization

The coordinates were normalized to a range between 0 and 100 as it was creating a smaller iamge size in a 1000 X 1000 matrix and then multiplies by the matrix size. This ensures that the data fits within the bounds of the matrix size. The normalization is performed using the following formula:

$$x_{\text{normalized}} = \frac{x - \text{range}_{\text{min}}}{\text{range}_{\text{max}} - \text{range}_{\text{min}}} \times (\text{matrix\_size} - 1) \tag{1}$$

$$y_{\text{normalized}} = \frac{y - \text{range}_{\text{min}}}{\text{range}_{\text{max}} - \text{range}_{\text{min}}} \times (\text{matrix\_size} - 1) \tag{2}$$

where $\text{range}_{\text{min}} = 0$, $\text{range}_{\text{max}} = 100$, and $\text{matrix\_size} = 1000$.

2.4 Discretization

The normalized coordinates are rounded and converted to integer indices suitable for the matrix. This transforms continuous coordinates into discrete grid points:

$$x_{\text{discrete}} = \text{round}(x_{\text{normalized}}) \tag{3}$$

$$y_{\text{discrete}} = \text{round}(y_{\text{normalized}}) \tag{4}$$

3 Matrix Creation

3.1 Boolean Matrix

A 1000×1000 boolean matrix is created for the task which is initialized with `False`. The matrix is populated with `True` at the positions corresponding to the discrete coordinates:

$$\text{sparse\_matrix}[x_{\text{discrete}}, y_{\text{discrete}}] = \text{True} \tag{5}$$

4 Matrix Transformations

4.1 Rotation

The origin lies at the lower left. To rotate the matrix 90° clockwise, we can first flip the matrix vertically and then transpose it. This can be expressed as:

$$\text{rotated\_matrix} = \text{sparse\_matrix}[\text{::-1}, :]^T \tag{6}$$

where [::-1] reverses the rows of the matrix and ^{T} denotes the transpose operation.

4.2 Flipping

Now, to flip the original matrix vertically (top-to-bottom), we can reverse the columns and then transpose the result:

$$\text{flipped\_matrix} = \text{sparse\_matrix}[:, :: -1]^T \tag{7}$$

Here, [:, ::-1] reverses the columns and ^{T} does the transpose operation.

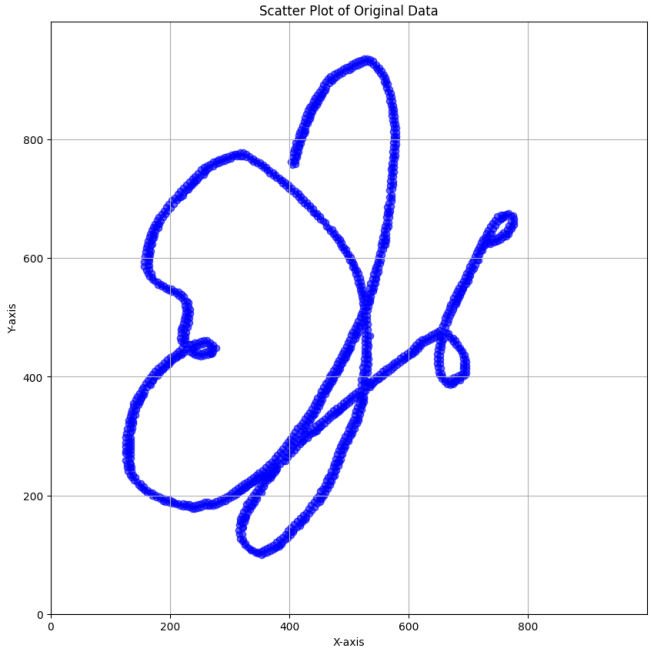
5 Visualization

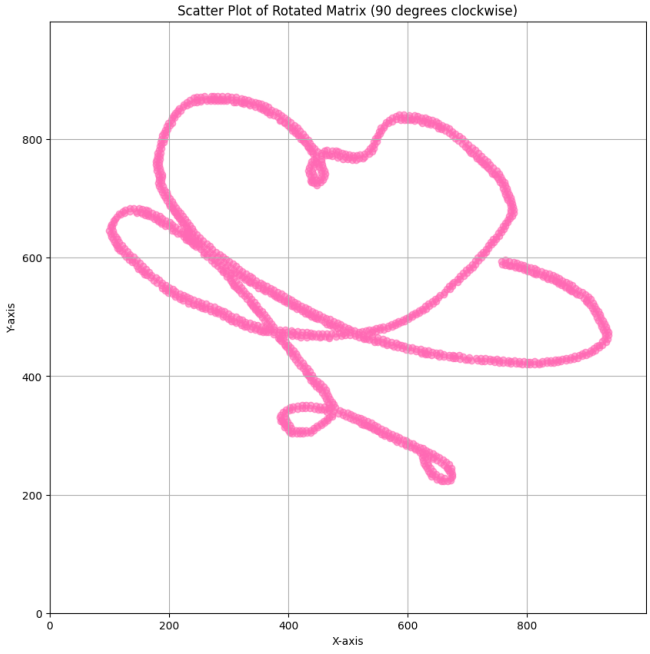
5.1 Matrix Visualization

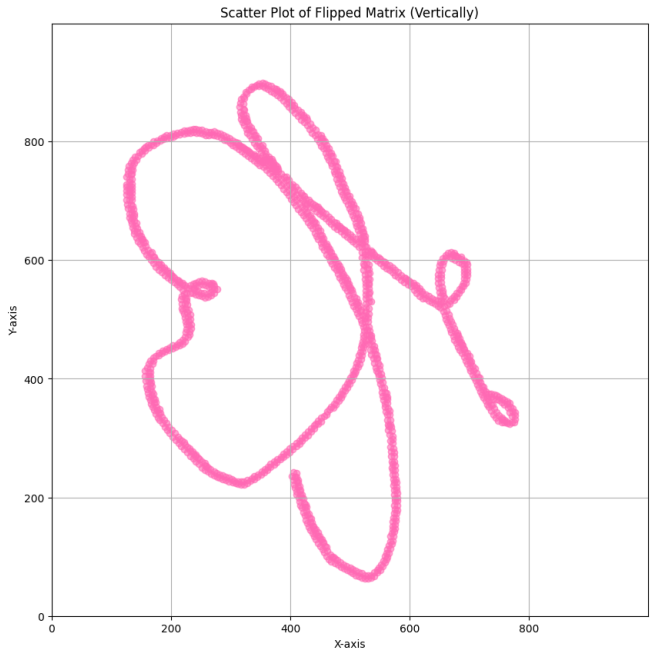
The matrices are visualized using `matplotlib` library. The boolean matrices are converted to integer matrices for visualization. Scatter plots are generated to show the distribution of data points. For the original, rotated, and flipped matrices, the scatter plots are created using:

```
plt.scatter(x_indices, y_indices, color='hotpink/blue', marker='o', alpha=0.6)
```

where x_{indices} and y_{indices} are the indices of `data` values in the matrices.


Scatter Plot of Original Data
Y-axis
X-axis
0
200
400
600
800


Scatter Plot of Rotated Matrix (90 degrees clockwise)
Y-axis
X-axis
0
200
400
600
800



6 Conclusion

The code uploaded processes and transforms data from a CSV file into various matrix forms, visualizing each step. The transformations include normalization, discretization, rotation, and flipping, with corresponding visualizations in both matrix and scatter plot forms.

7 References

- Numpy docs
- Matplotlib docs
- Webplotdigitizer
- StackOverflow
- Overleaf docs