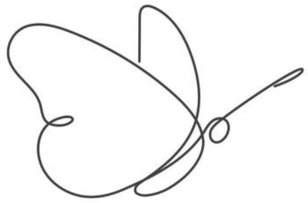
DA5401 Assignment #1

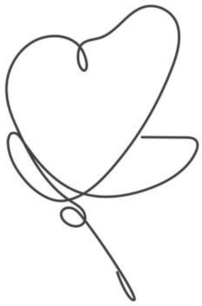
Welcome to the world of Data Science where we set our sail from Raw Data to Actionable Decisions. As an introductory exercise, we will venture into a Data Acquisition task, where we will convert a real world system into a dataset. As the next step, we shall perform some algebraic operations to visualize the result of such transformations. In this process, you will learn about using tools to create datasets, perform basic data engineering tasks, run matrix operators on the data, and finally use tools to visualize the outcomes. Let’s roll!

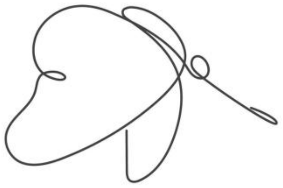
Real world system

Consider drawing a line sketch of your favorite object. The following is such an example sketch of a butterfly (1st image). You are expected to create the 2nd image by rotating the 1st clockwise by 90 degrees. Likewise, flip the 1st image horizontally to create the 3rd image. Of course, you can use a standard image processing tool to do this in a jiffy, but you are expected to do this programmatically through matrix operations.

Remember, you will draw a sketch yourself in a piece of white paper. Take a photo of that sketch using your phone. Ensure that your image is (cropped to a) square for mathematical ease. Upon completing the steps, you will get your own version of the 1st image.

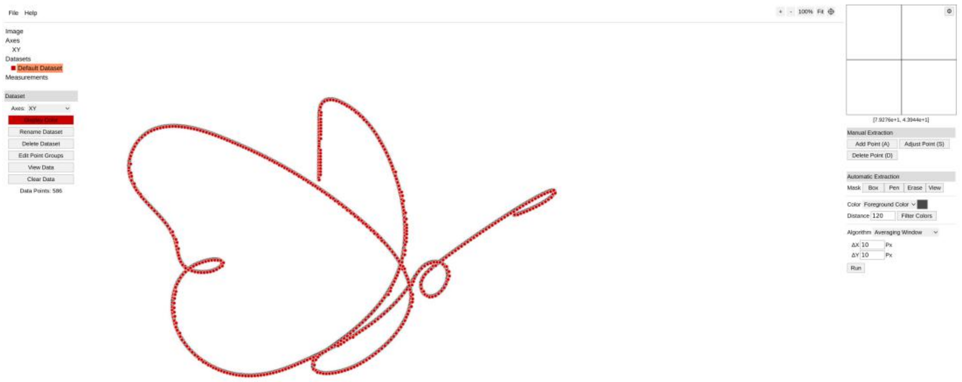






Data Acquisition [20 points]

Our goal is to use a tool, such as (but not limited to), https://wpd.starrydata2.org/ to convert the line-sketch into a table of x-y coordinates. We will call this as the 2d dataset. You may have to see the tutorial videos on how to convert an image to a dataset. Trust me, it shouldn’t take more than 10 mins to figure out the process. Upon conversion, you should see something like below in the tool.



Axes: XY

Data Points: 586

[7.9276e+1, 4.3944e+1]

Color: Foreground Color

Distance: 120

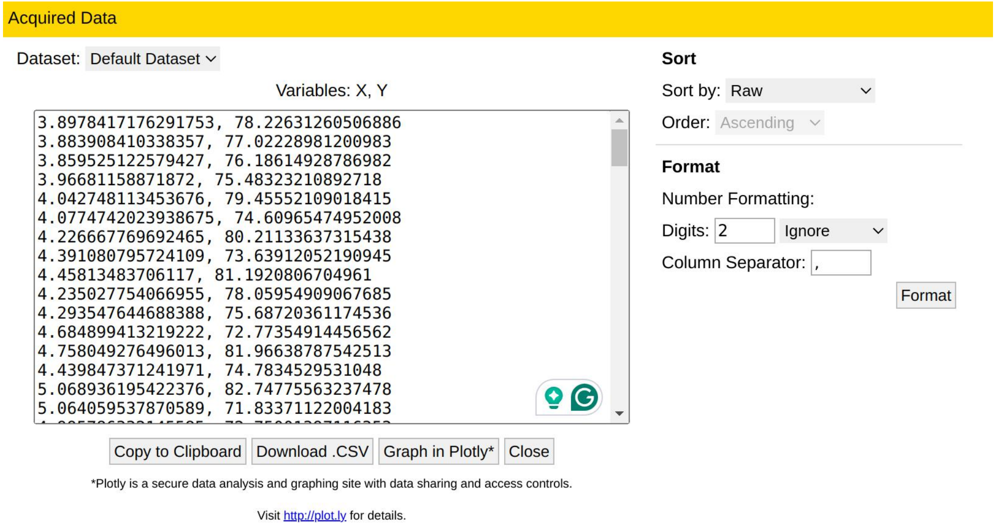
Algorithm: Averaging Window

ΔX 10 Px

ΔY 10 Px

The tool will ask you to choose fiducial points {X1, X2, Y1, Y2} to fix the X and Y axes. Also you should set the value of X and Y appropriately to define the ranges of X and Y axes. In the example, I chose to use X = [1,100] and Y = [1, 100]. My X1 and X2 were 10 and 90, likewise my Y1 and Y2 were 5 and 80 depending on where I placed the fiducials points on the imaginery X and Y axes. You can choose your range of interest. Once you set everything, choose the “color” of the line sketch to match your drawing color and then hit “Run” for the data conversion task to complete.

The red dots, that you see on the above image are the extracted data points. Upon clicking the “View Data” on the left pane, you should see a popup window like below. Clicking on the “Download CSV” should get you the dataset to move further.



Acquired Data

Dataset: Default Dataset

Variables: X, Y

```
3.8978417176291753, 78.22631260506886
3.883908410338357, 77.02228981200983
3.859525122579427, 76.18614928786982
3.96681158871872, 75.48323210892718
4.042748113453676, 79.45552109018415
4.0774742023938675, 74.60965474952008
4.226667769692465, 80.21133637315438
4.391080795724109, 73.63912052190945
4.45813483706117, 81.1920806704961
4.235027754066955, 78.05954909067685
4.293547644688388, 75.68720361174536
4.684899413219222, 72.77354914456562
4.758049276496013, 81.96638787542513
4.439847371241971, 74.7834529531048
5.068936195422376, 82.74775563237478
5.064059537870589, 71.83371122004183
```

Sort

Sort by: Raw

Order: Ascending

Format

Number Formatting:

Digits: 2 Ignore

Column Separator: ,

*Plotly is a secure data analysis and graphing site with data sharing and access controls.

Visit http://plot.ly for details.

Data Cleansing & Loading [10 points]

Load the CSV into a pandas dataframe and perform any kind of cleansing to have only valid data points in your dataframe. You may then transform the dataframe into a 2D sparse matrix by discretization. Remember that the dataframe consists only the coordinates of the red points, but we want to represent the entire image as a 2D numpy matrix, which will be very sparse. If you consider your dataset to have the 0-100 range, you are expected to construct a 100x100 matrix of the image. When the matrix size is smaller because of hard discretization, the image quality may suffers. So, you may consider discretizing your data to a 1000x1000 boolean matrix. Larger the better!

Transformation [10 points]

Figure out the matrix multiplication operations that would get your matrix rotated by 90 degrees to generate the 2nd image. Likewise, figure out the matrix operation that will flip the matrix horizontally to get the 3rd image. Write the code using numpy operators on the numpy matrix to perform the transformations. After the transformations are completed, you are expected to convert the sparse matrices into their respective X-Y coordinates.

Visualization [10 points]

Using X-Y scatterplot method of any library (matplotlib or seaborn), generate the scatter plot to visualize the 2nd and 3rd images from only the converted X-Y coordinates. If your images turn out to be coarser, consider increasing the size of the matrix representation.

Note: You should use not use OpenCV or equivalents in your code.

Data Analytics is Fun!!