Exploring Multimodal Inputs for Neural Network Inverse Kinematics: Constrained Workspace Optimization

Saatvik Tammishetty
B.Tech, Computer Science and Engineering (211112279)

Under the Guidance of
Dr. Vijay Bhaskar Semwal

Contents

Introduction

Analyzing a robotic arm involves determining the orientation and positioning of objects within a specified domain. Describing the motion of a robot arm involves specifying its orientation and position within a coordinate system, commonly using Cartesian coordinates with perpendicular X and Y axes.

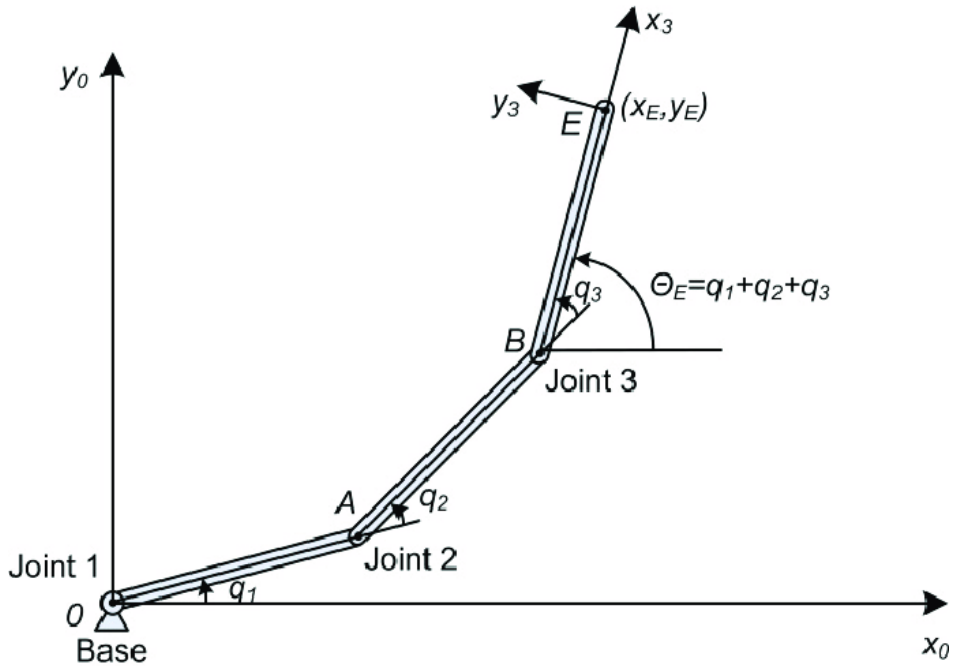

Figure 1: 3-Link Manipulator in a Cartesian Coordinate Plane [1]

Motivation

Automation becomes crucial in situations where humans are required to operate in environments filled with toxins and pollutants, posing significant risks to health.

Figure 2: Polluted Environment [2]

Figure 3: Underwater Excavations [4]

Figure 4: Radioactive Facility [6]

Motivation

The need for automation arises due to the potential for severe health consequences in such hazardous conditions.

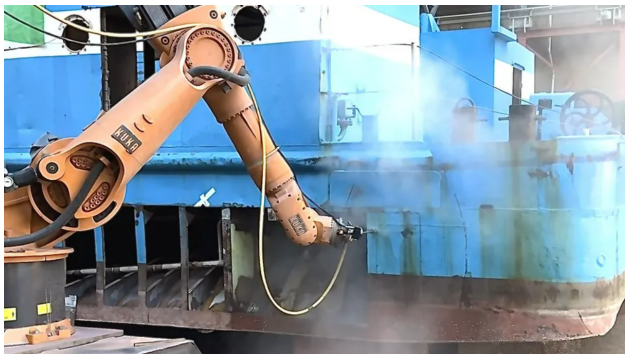

Figure 5: Working in Toxic Fumes [3]

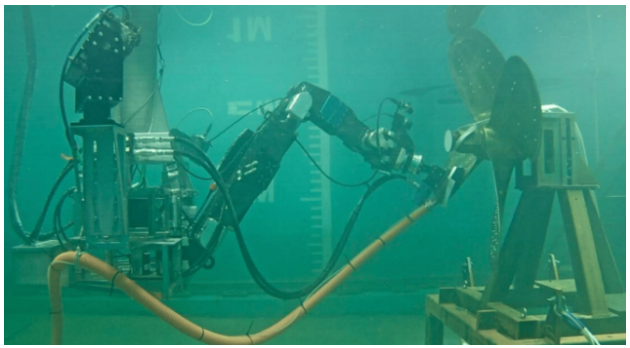

Figure 6: Hydraulic Manipulator Arm [5]

Figure 7: Safe Handling of Nuclear Residue [7]

Terminology

Link Lengths (l1, l2, l3): Physical lengths of the robot's links connecting the joints.

Joint Angles ($\theta_1, \theta_2, \theta_3$): Variables representing the angular positions of each joint in the manipulator.

End-Effector: The tool or device attached to the last link of the manipulator that interacts with the environment.

Workspace: The spatial region within which the end-effector of a robot manipulator can move or operate.

Workspace Analysis: Evaluating the reachable and unreachable regions in the workspace.

Joint Limits: The constraints on the range of motion for each joint.

Degrees of Freedom (DOF): The number of independent movements or rotations a robot manipulator can perform.

Test Model

A 3-Link Manipulator consists of three interconnected links or segments joined by joints, allowing relative motion between them. It typically consists of three joints, one for each link connection. We have considered a simple 3-Link Manipulator as our test model.

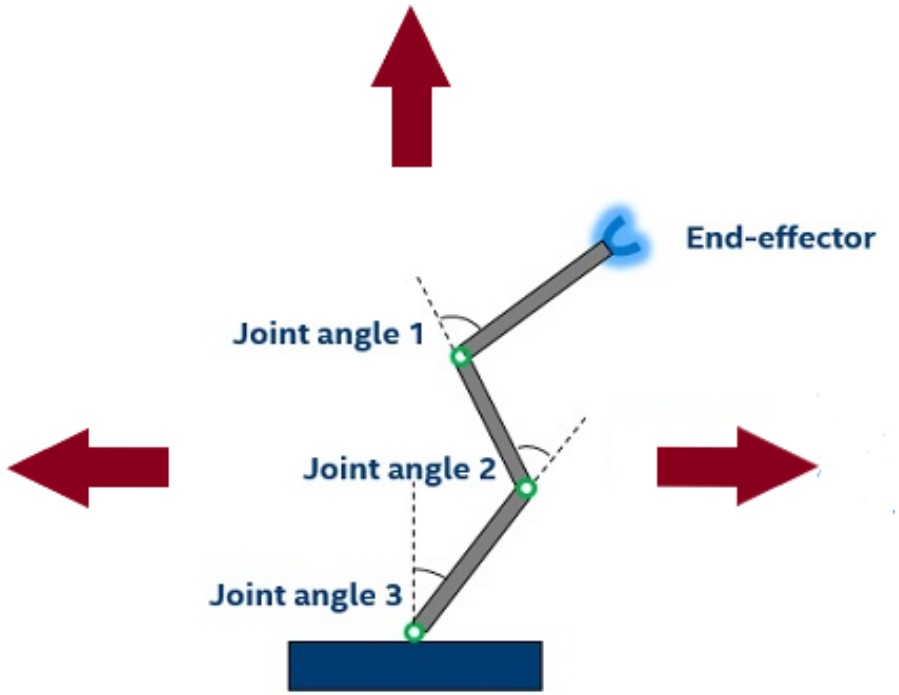

Figure 8: A Simple 3-Link Manipulator [8]

Kinematics

Kinematics in robotics is a fundamental aspect that deals with the study of motion of robotic systems, both in terms of position and orientation. It encompasses two main components: forward kinematics and inverse kinematics. Understanding forward and inverse kinematics is essential for designing, controlling, and programming robotic systems. They play a pivotal role in various applications, from industrial robotic arms to humanoid robots, enabling precise motion planning and execution.

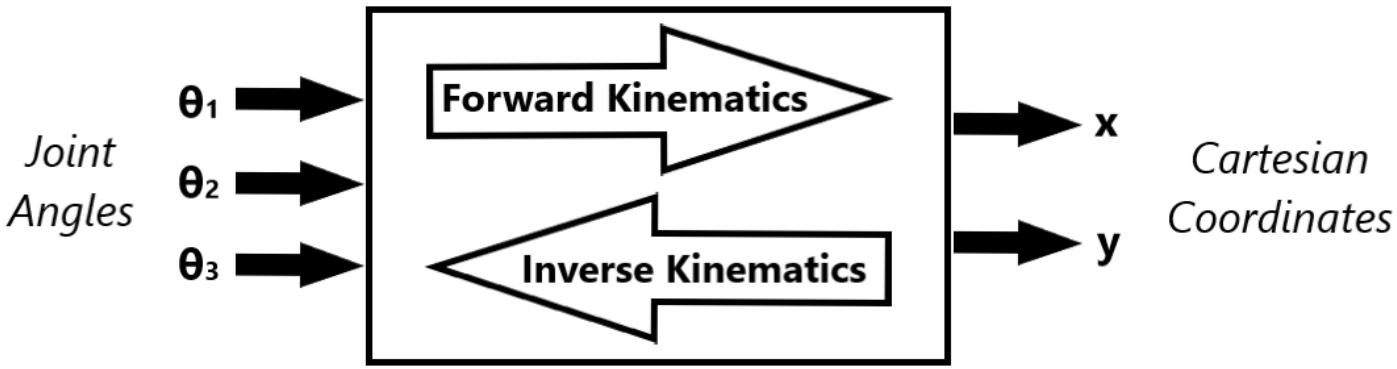

Figure 9: Relationship between Forward and Inverse Kinematics

Forward Kinematics

Forward kinematics addresses the relationship between the individual joints of a robot manipulator and the position and orientation of the end-effector. It calculates the overall motion of the robot from the input joint values, providing a mapping from joint space to Cartesian space.

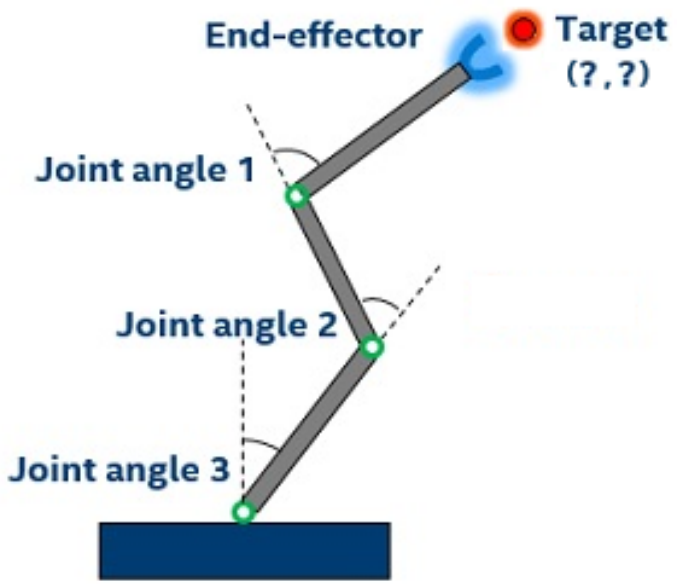

Figure 10: Forward Kinematics [8]

Forward Kinematic Equations

The equations derived from Forward Kinematics help understand how a robot moves and where its end-effector will point for given joint configurations. Consider a 3-link manipulator having joint angles $(\theta_1, \theta_2, \theta_3)$. The equations of Forward Kinematics are as follows.

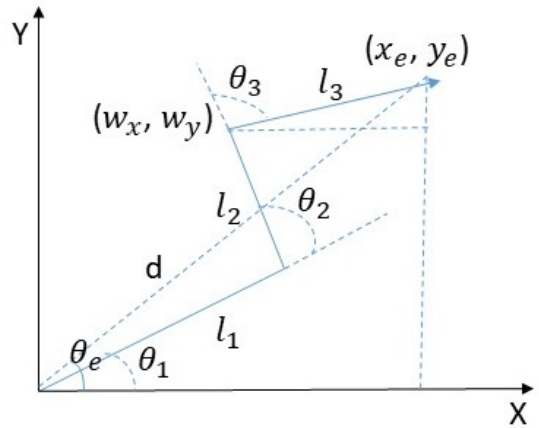

Figure 11: Orientation of the 3-Link Manipulator [12]

$$x_e = l_1 cos(\theta_1) + l_2 cos(\theta_1 + \theta_2) + l_3 cos(\theta_1 + \theta_2 + \theta_3)$$
$$y_e = l_1 sin(\theta_1) + l_2 sin(\theta_1 + \theta_2) + l_3 sin(\theta_1 + \theta_2 + \theta_3)$$
$$\theta = \theta_1 + \theta_2 + \theta_3$$

Inverse Kinematics

Inverse kinematics involves determining the joint variables for a given position of end effector. The resolution of inverse kinematics is crucial for calculating the precise position and orientation of the end effector of a robot manipulator to accomplish its task.

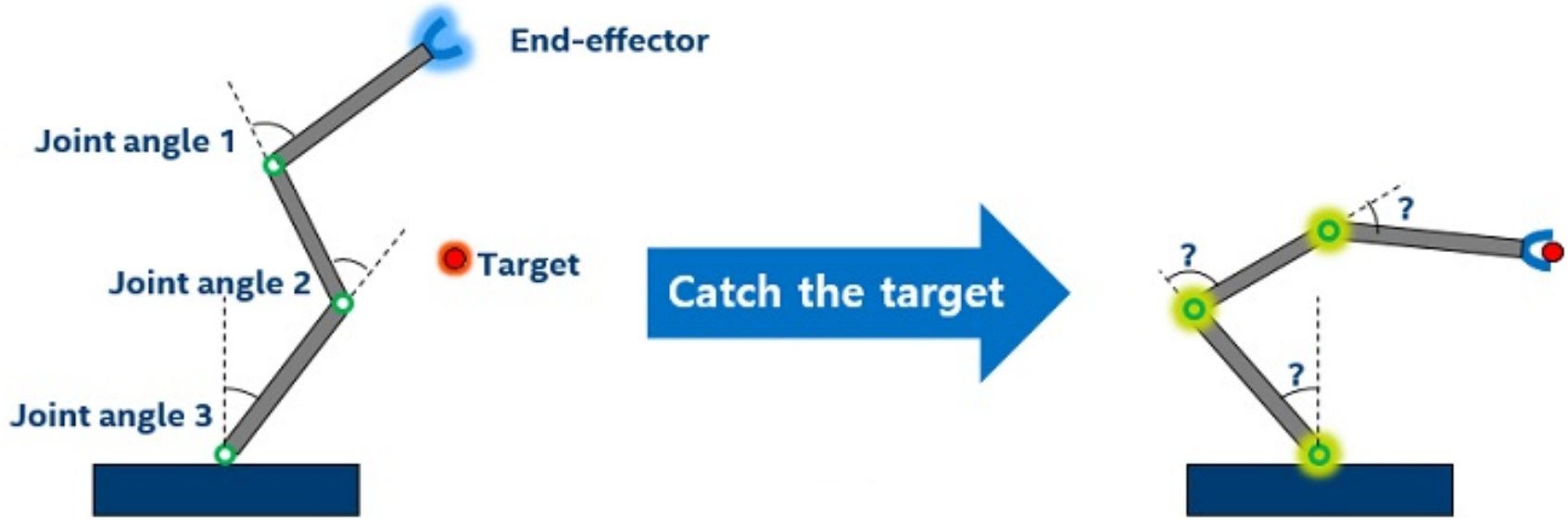

Figure 12: Inverse Kinematics [8]

Problems in Inverse Kinematics

The primary challenge associated with inverse kinematic formulations lies in computational and mathematical complexity due to higher-degree polynomial half-tangent equations, which do not guarantee closed-form solutions.

A number of joint angle combinations may be available for a given target point.

If the target point requires a near-singular robot configuration, joint angle estimation is difficult

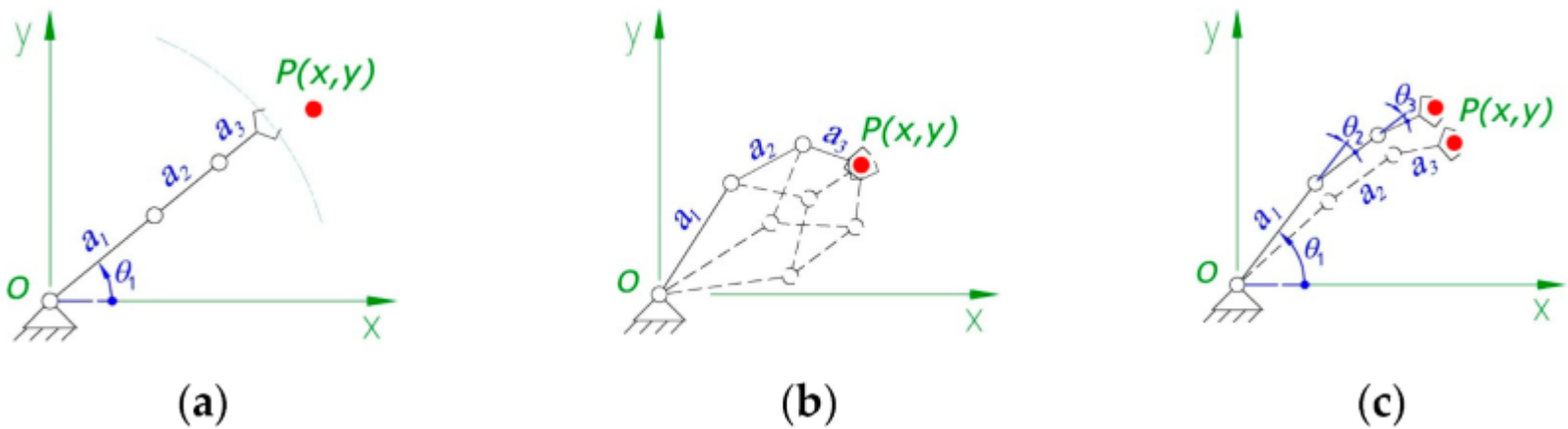

Figure 13: Problems in inverse kinematics solutions [9]

Applications of Inverse Kinematics

Inverse Kinematics plays a significant role in gait analysis. Gait analysis involves studying human walking patterns and understanding the underlying joint movements.

They also play a crucial role in automating various industrial processes, resulting in better productivity.

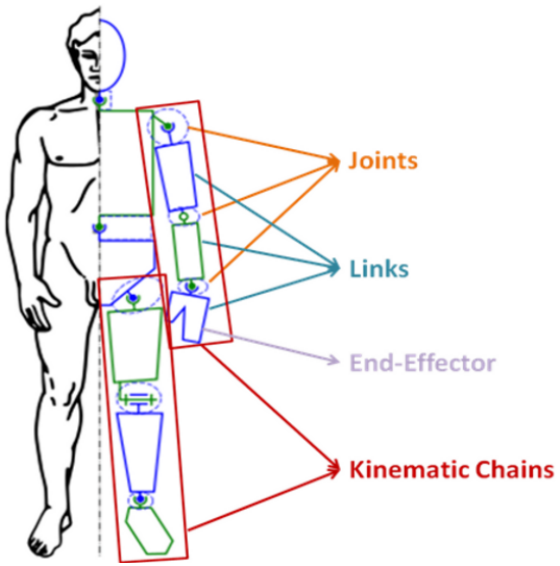

Figure 14: Human Analogy [10]

Figure 15: Industrial Automation [11]

Methods of Solving Inverse Kinematics

Analytical Approach

Jacobian Transpose

Pseudo - Inverse Jacobian

Levenberg - Marquardt Damped Least Squares

Quasi - Newton and Conjugate Gradient

FABRIK (Forward And Backwards Reaching IK)

Neural Networks

Cyclic Coordinate Descent

Genetic Algorithm

Particle Swarm Optimization

Analytical Approach for Inverse Kinematics

Inverse Kinematics solution can be obtained by solving the equations of Forward Kinematics for the joint angles.

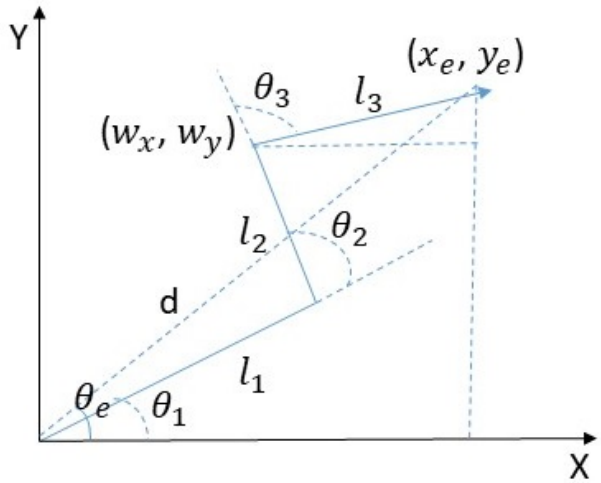

Figure 16: Solution to 3-Link Manipulator [12]

$$x_e = l_1 cos(\theta_1) + l_2 cos(\theta_1 + \theta_2) + l_3 cos(\theta_1 + \theta_2 + \theta_3)$$

$$y_e = l_1 sin(\theta_1) + l_2 sin(\theta_1 + \theta_2) + l_3 sin(\theta_1 + \theta_2 + \theta_3)$$

$$\theta_e = \theta_1 + \theta_2 + \theta_3$$

Analytical Approach for Inverse Kinematics

We can find out the position (w_x, w_y) from simple trigonometric equations as follows:

$$w_x = x_e - l_3 cos(\theta_e)$$

$$w_y = y_e - l_3 sin(\theta_e)$$

Now, using the cosine rule,

$$w_x^2 + w_y^2 = l_1^2 + l_2^2 - 2 l_1 l_2 cos(\theta_2)$$

$$cos(\theta_2) = (w_x^2 + w_y^2 - l_1^2 - l_2^2)/2 l_1 l_2$$

Using Inverse Cosine Function

$$\theta_2 = cos^{-1}((w_x^2 + w_y^2 - l_1^2 - l_2^2)/2 l_1 l_2)$$

From Figure 16, we can also infer the position (w_x, w_y) as shown below :

$$w_x = l_1 cos(\theta_1) + l_2 cos(\theta_1 + \theta_2)$$

$$w_y = l_1 sin(\theta_1) + l_2 sin(\theta_1 + \theta_2)$$

Analytical Approach for Inverse Kinematics

These equations can be rewritten as follows :

$$w_x = (l_1 + l_2 cos(\theta_2))cos(\theta_1) - l_2 sin(\theta_1)sin(\theta_2)$$

$$w_y = (l_1 + l_2 cos(\theta_2))sin(\theta_1) - l_2 cos(\theta_1)sin(\theta_2)$$

On solving for $cos(\theta_1), sin(\theta_1)$ we get

$$cos(\theta_1) = ((l_1 + l_2 cos(\theta_2))x_2 + l_2 sin(\theta_2)y_w)/(x_w^2 + y_w^2)$$

$$sin(\theta_1) = ((l_1 + l_2 cos(\theta_2))y_2 - l_2 sin(\theta_2)x_w)/(x_w^2 + y_w^2)$$

Now, θ_1 can be calculated as shown below

$$tan(\theta_1) = sin(\theta_1)/cos(\theta_1)$$

$$\theta_1 = tan^{-1}(sin(\theta_1)/cos(\theta_1))$$

Analytical Approach for Inverse Kinematics

From the equations of Forward Kinematics,

$$\theta_e = \theta_1 + \theta_2 + \theta_3$$

Also,

$$theta_e = tan^{-1}(y_e/x_e)$$

Substituting the values of θ_e, θ_1 and θ_2 in the above equation,

$$\theta_3 = tan^{-1}(y_e/x_e) - \theta_1 - \theta_2$$

The results obtained are almost error free and very accurate. But, the main limitation with this model is that it requires heavy calculation and explicit programming for every type of manipulator.

Neural Network Approach

Neural networks can be used to solve any kind of equations, including the complex nonlinear equations of Inverse Kinematics. In our case we have used a feed-forward network consists of 3 inputs, two hidden layers of 100 neurons each and 3 neurons in the output layer.

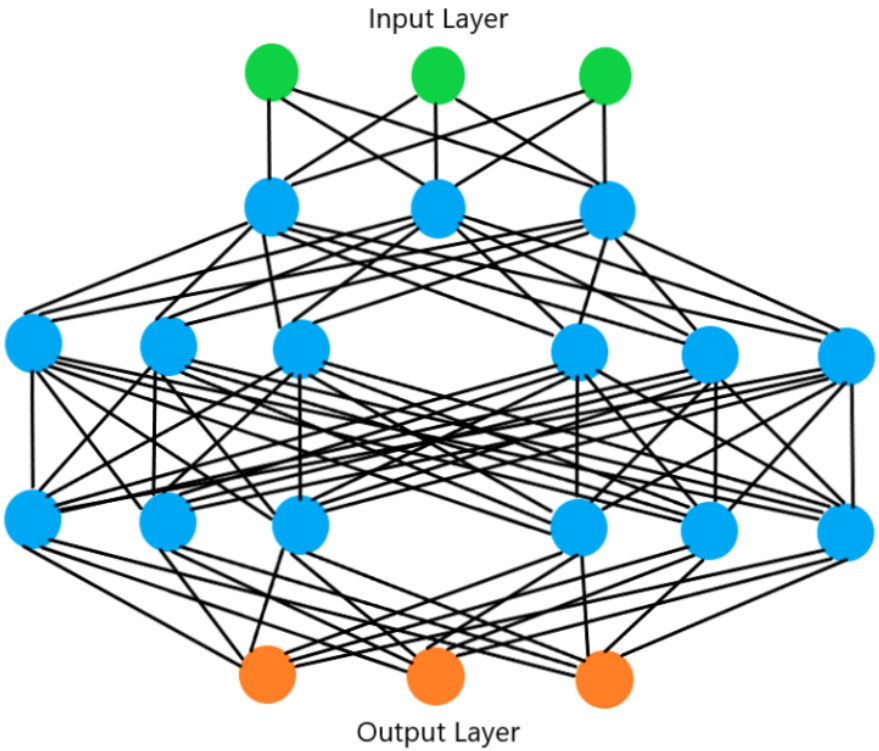

Figure 17: Artificial Neural Network

Preparation of Dataset

Training an Artificial Neural Network Model requires data. Here, we will train our neural network by creating our own data set from the equations of Forward Kinematics. In our case we have created 150000 samples. This training data can be represented on the Cartesian Coordinate Space as shown below.

x_scaled	y_scaled	theta_scaled	q1_estimated	q2_estimated	q3_estimated
11.7	-0.94	-9.74	1.27	-2.44	1
17.96	1.76	-38.39	0.58	-0.25	-1
2.29	-10.59	-107.14	0.41	-2.28	0
5.54	0.62	-36.1	2.82	-2.45	-1
6.31	6.66	-59.01	1.75	-0.78	-2
12.68	-0.41	-5.73	1.2	-2.3	1
-3.69	17.57	114.59	2.28	-1.28	1
2.59	-8.72	-79.07	0.44	-2.82	1
-11.68	17.35	120.32	2.29	-0.19	0
0.82	19.21	112.87	1.63	-0.66	1
15.12	4.08	-11.46	1.37	-1.57	0
10.57	-1.08	-34.95	1.7	-2.31	0
7.2	17.85	93.97	1.27	-0.63	1
16.11	12.73	29.79	0.97	-0.45	0
11.02	17.52	50.99	1.25	-0.36	0
-1.67	-5.02	-138.66	1.2	-2.62	-1
3.7	-3.86	-68.18	2.54	-2.73	-1
11.85	16.9	47.56	1.22	-0.39	0

Figure 18: Model Accuracy

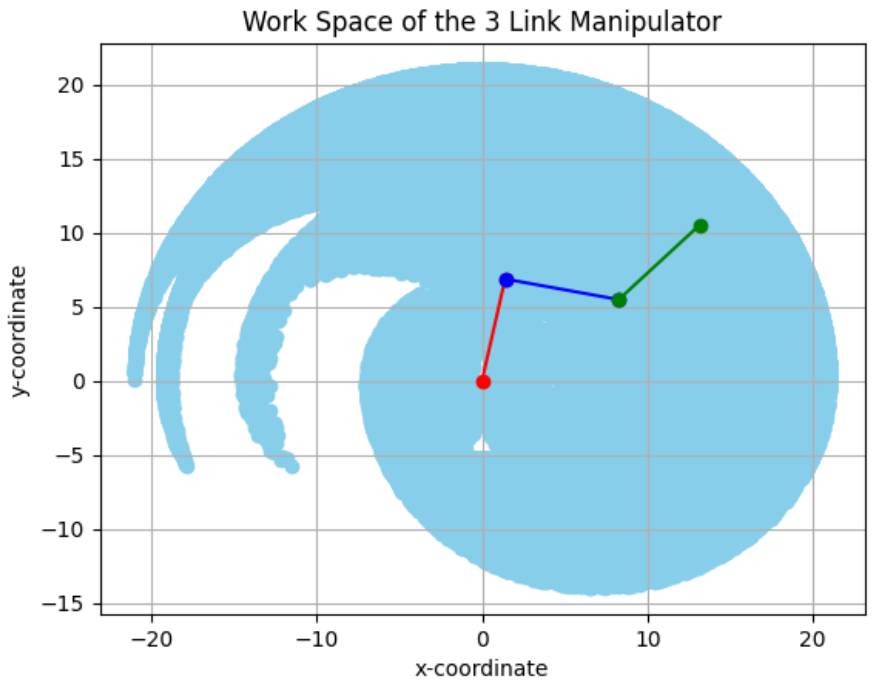

Figure 19: Work Space

Defining the Target Work Space

The target search space will be confined to a rectangular area, which is a part of the actual work space. This means that the final plot generated by our model must be confined to this region by necessary translation and scaling operations. Thus, we can ensures that the output plot will always lie within the Work Space.

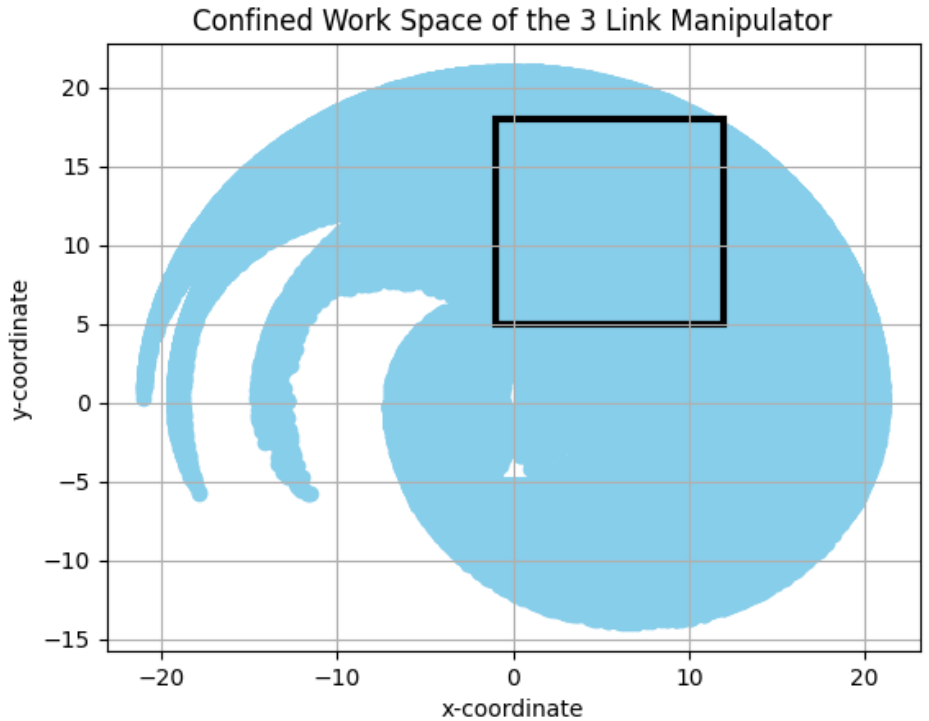

Figure 20: Confined Work Space

Performance Analysis of the Neural Network Model

The performance of the Neural Network Model can be analysed based on its Accuracy and Loss. The Accuracy of the model is around 99% while the Model Loss is around 0.0038.

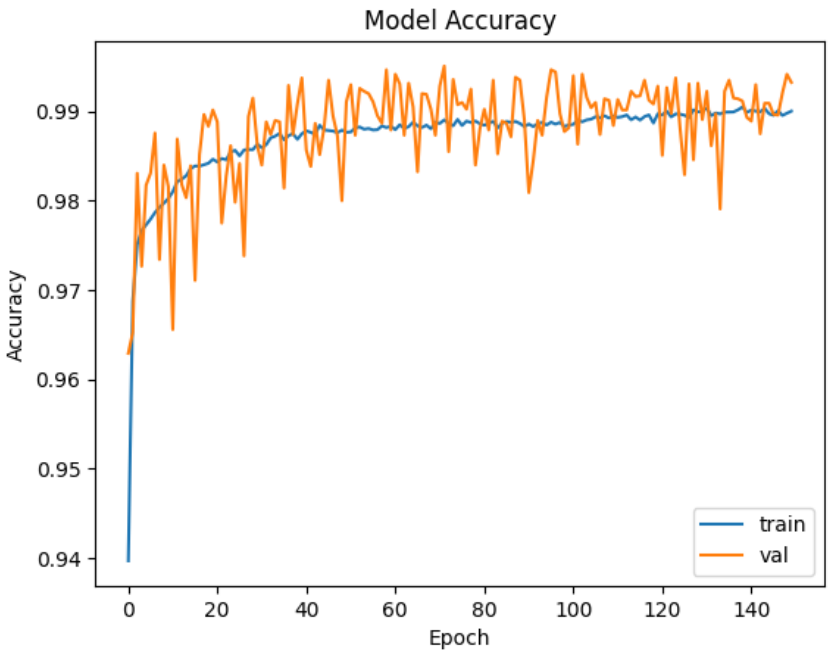

Figure 21: Model Accuracy

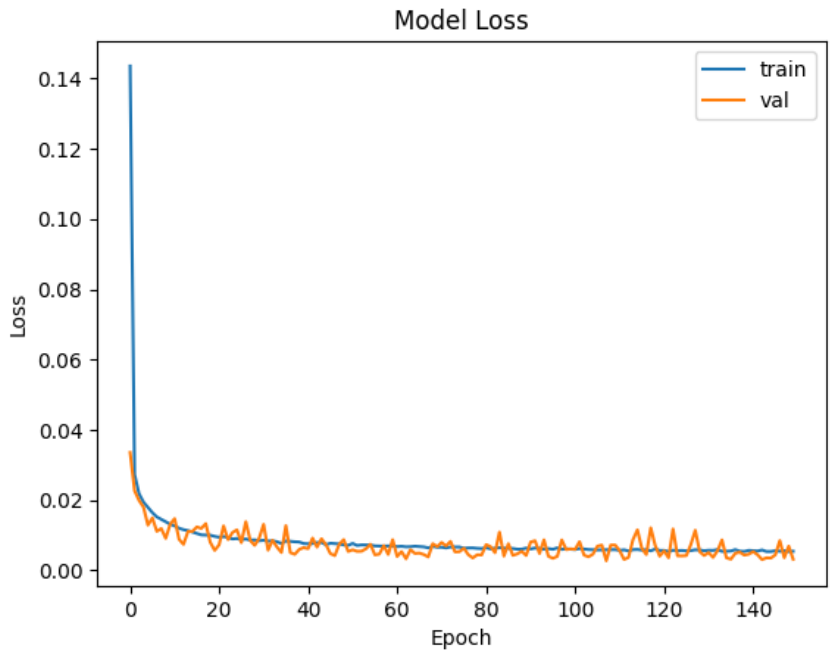

Figure 22: Model Loss

Experiments Performed

Multimodality of a model refers to the ability to handle different kinds of data seamlessly. This model may get the input in various forms, such as CSV file containing the numerical coordinates (x, y, theta), image, or even text input. This versatility allows the model to adapt to diverse sources of information, making it more flexible and applicable in different scenarios.

Figure 23: CSV File

Figure 24: PNG/JPG Image

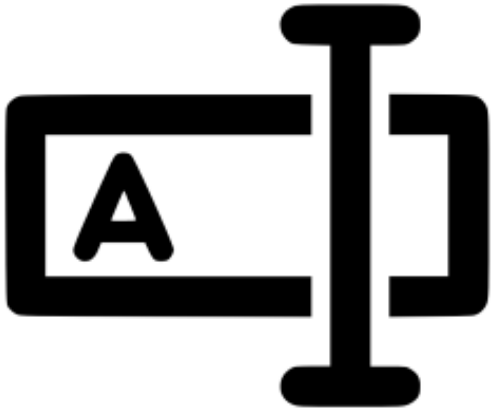

Figure 25: Text Input

Plot Generated from a CSV File of Coordinates

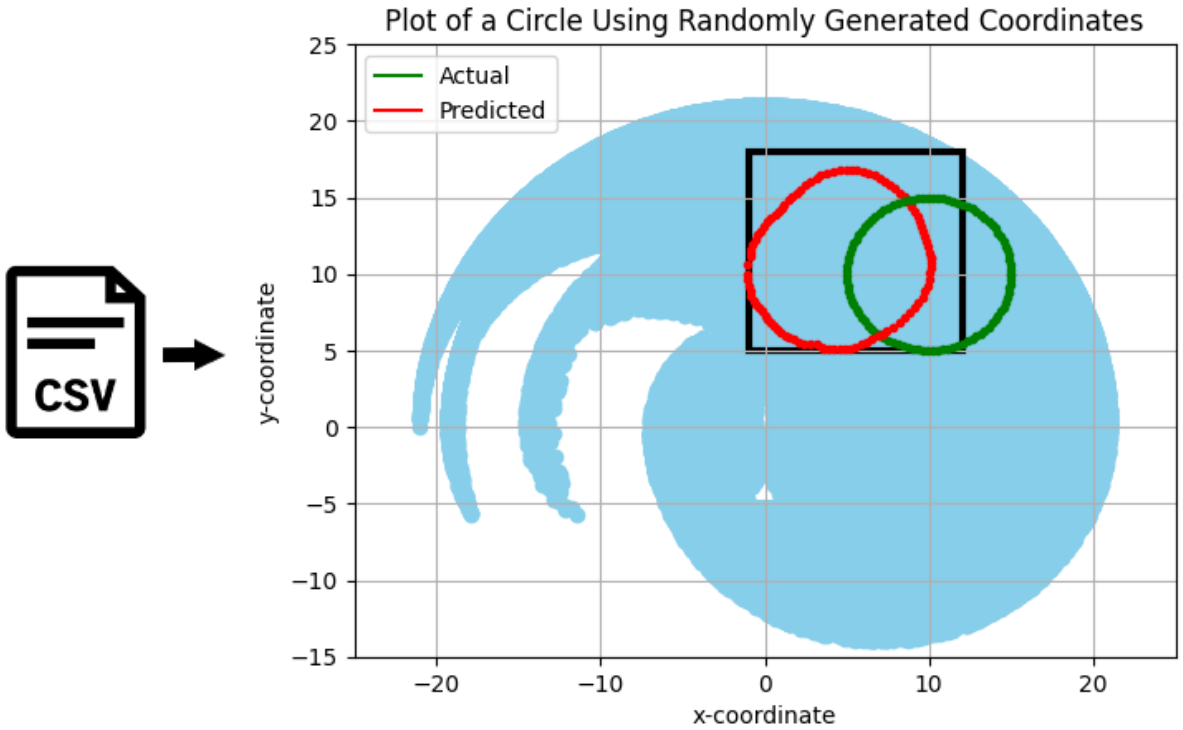

Figure 26: Plot of a Circle

Plot Generated from a CSV File of Coordinates

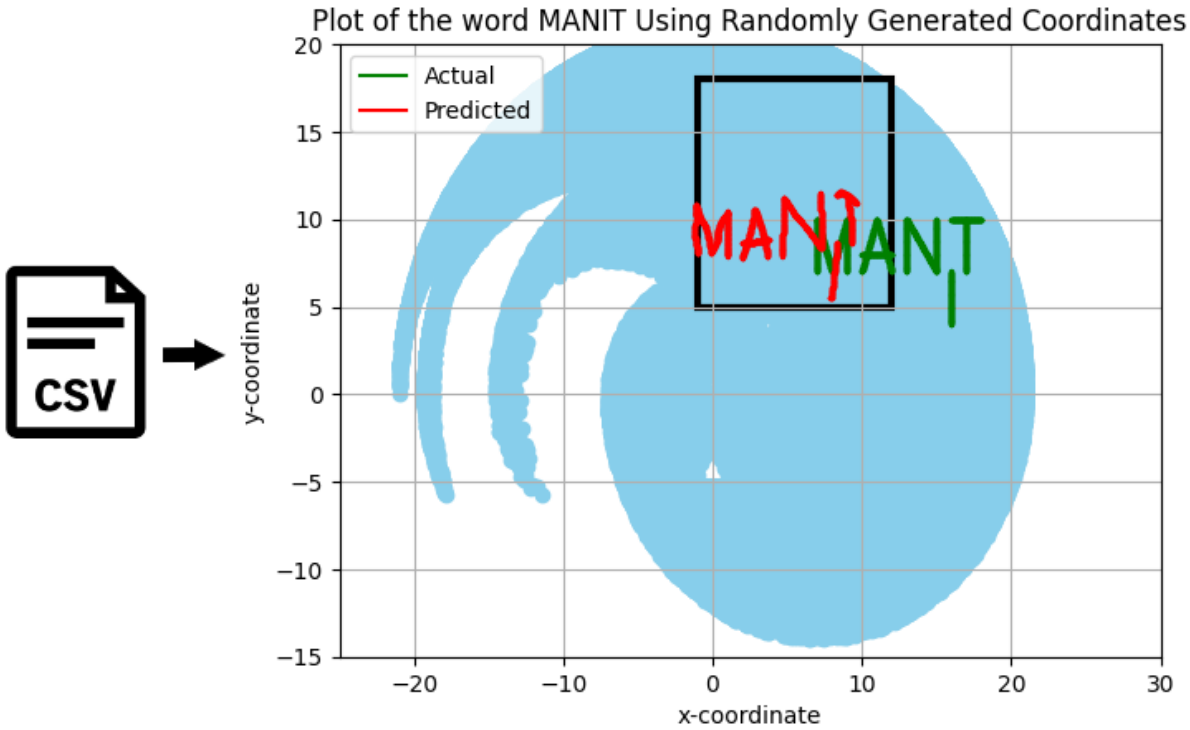

Figure 27: Plot of the word "MANIT"

Plot Generated from an Input Image

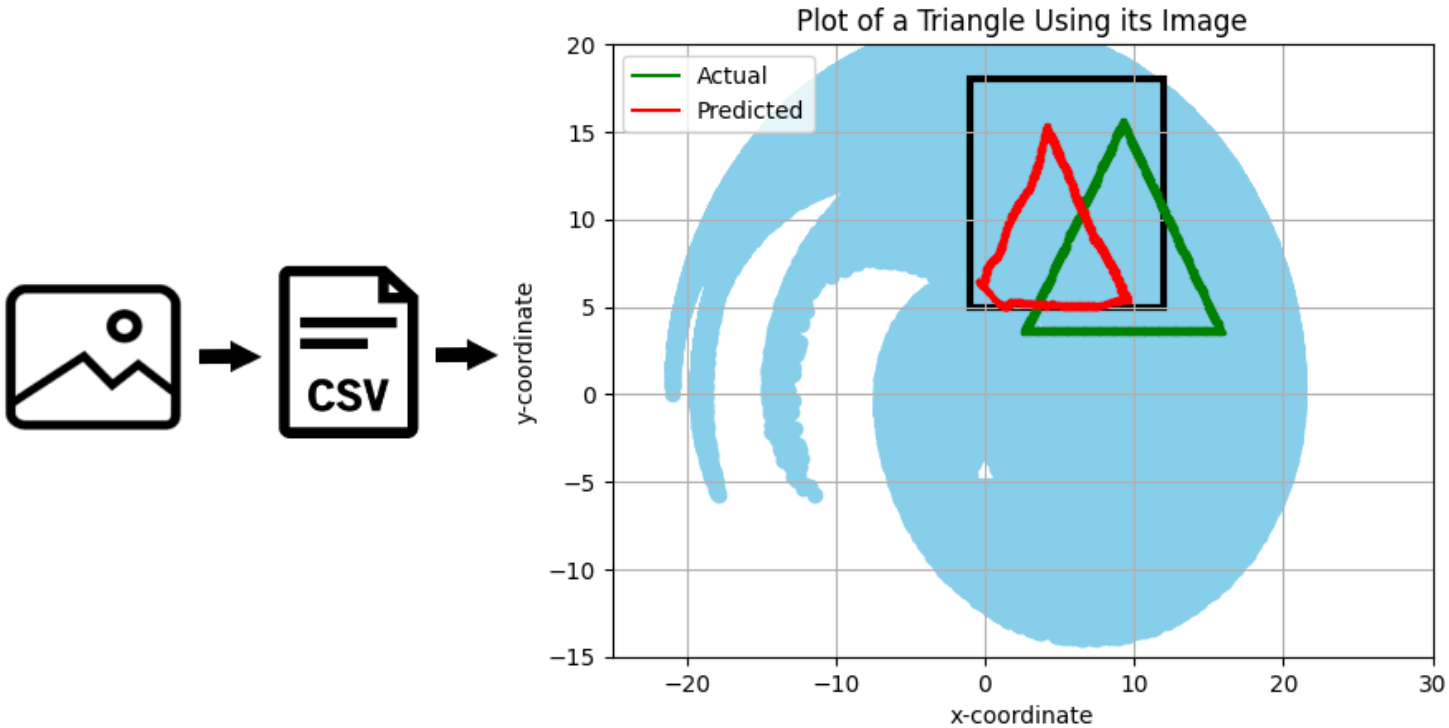

Figure 28: Plot of a Triangle

Plot Generated from an Input Image

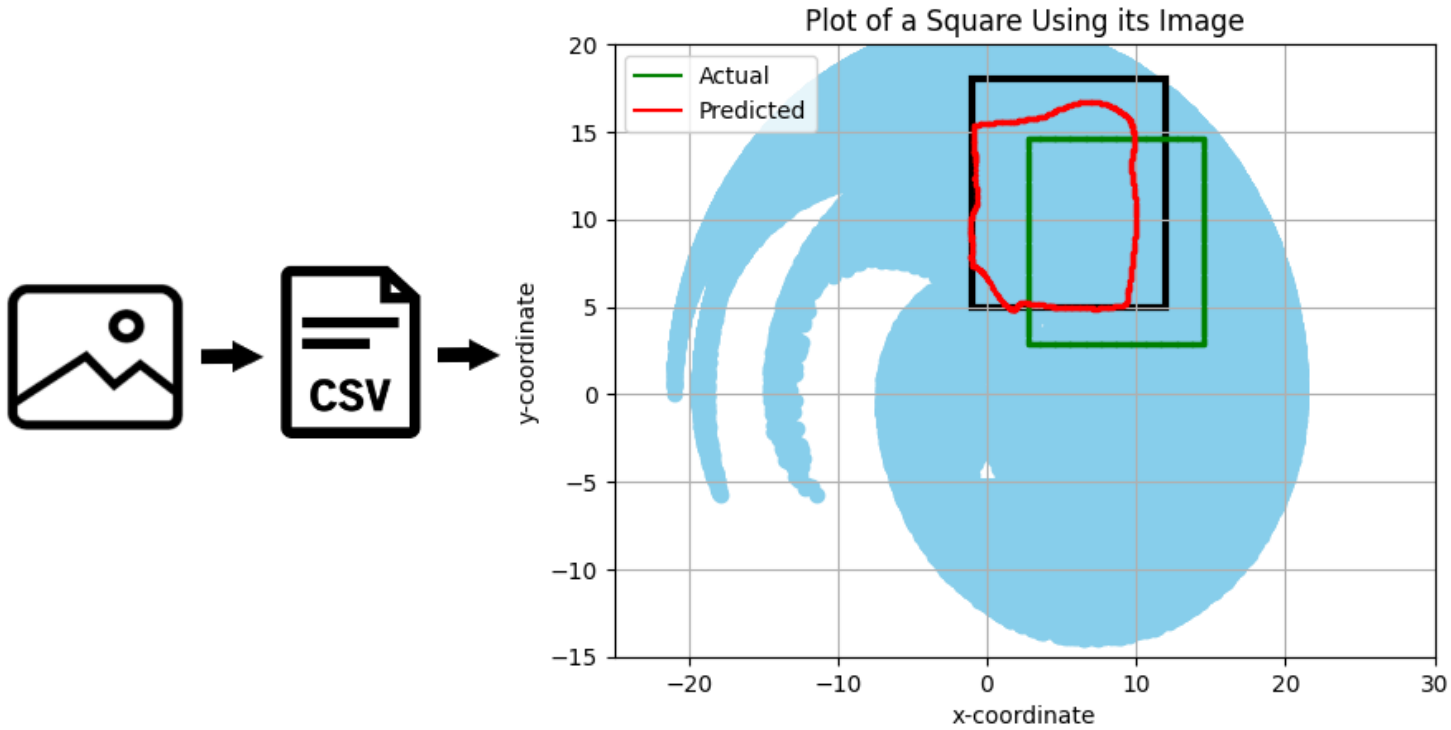

Figure 29: Plot of a Square

Plot Generated from an Input Image

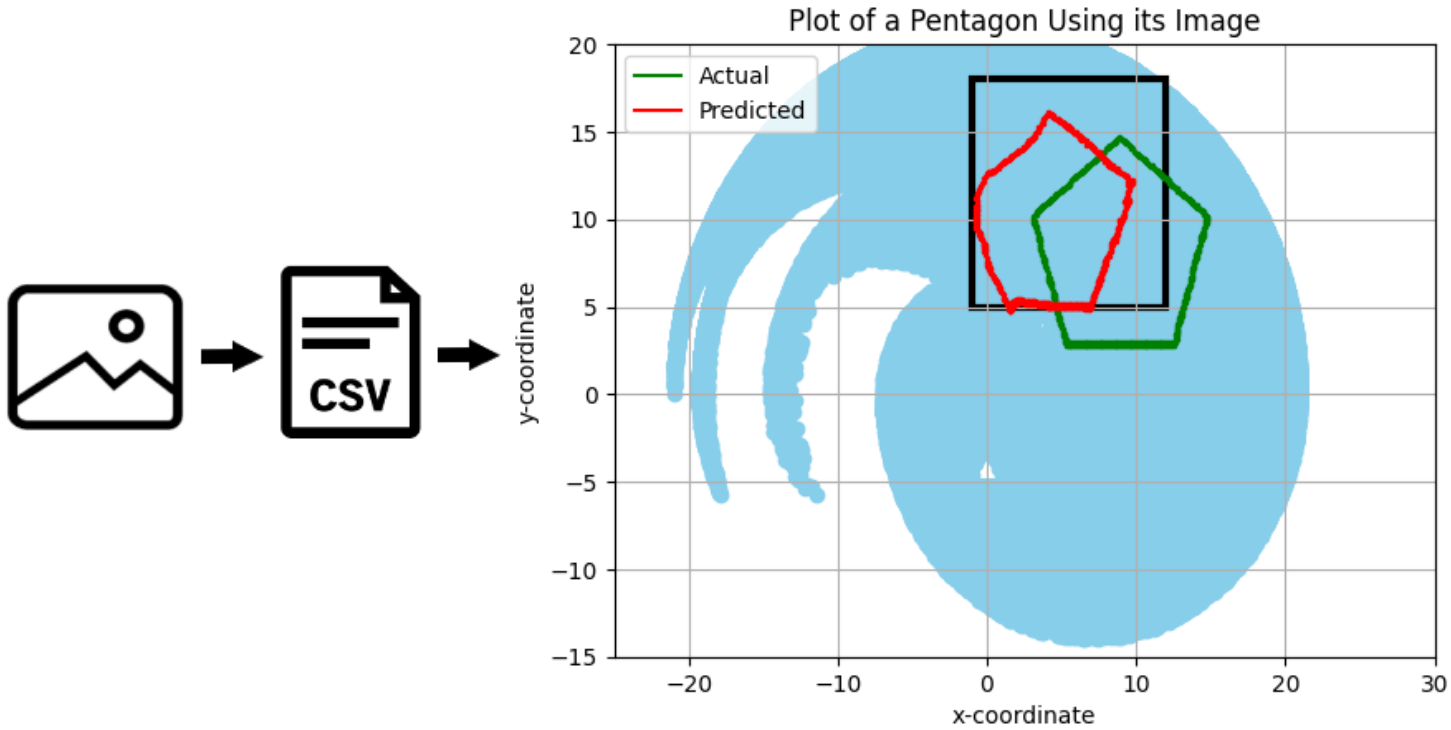

Figure 30: Plot of a Pentagon

Plot Generated from an Input Image

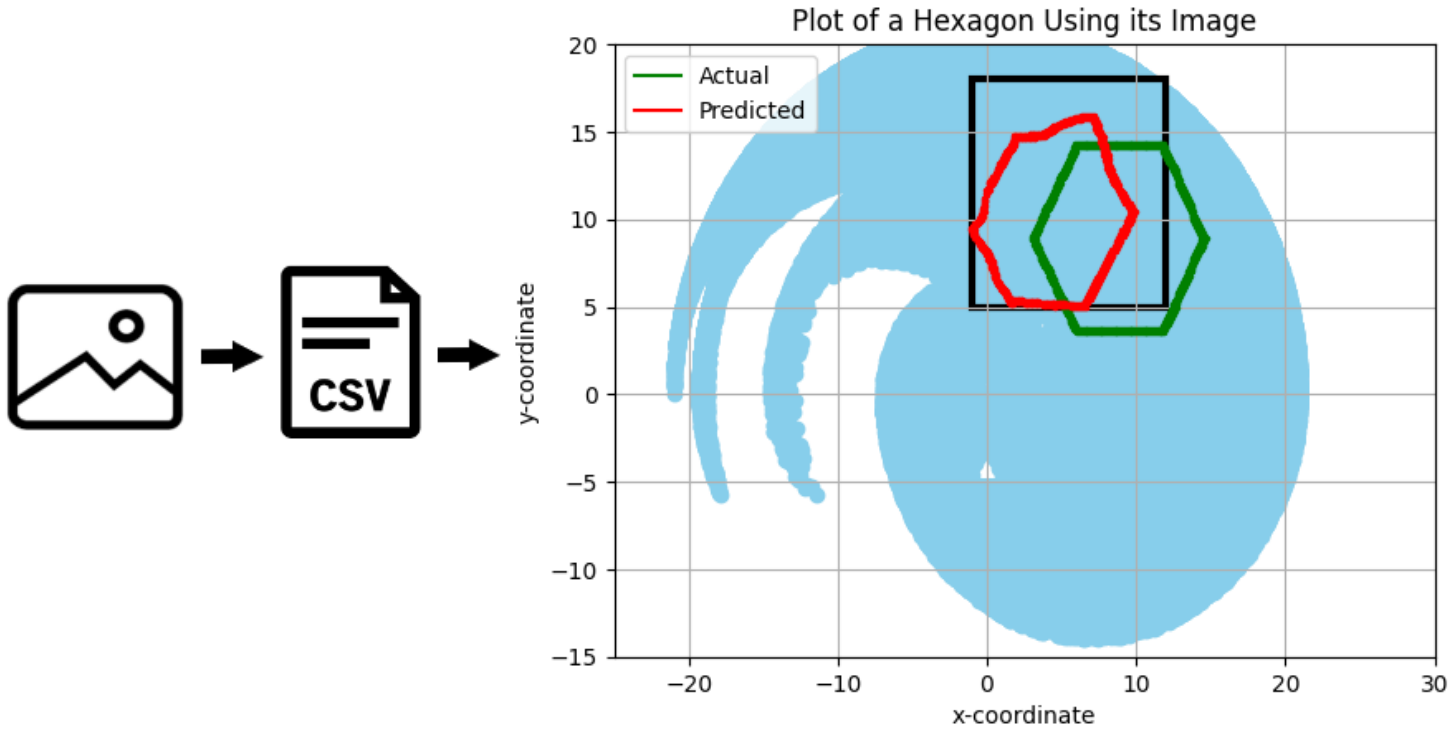

Figure 31: Plot of a Hexagon

Plot Generated from an Input Image

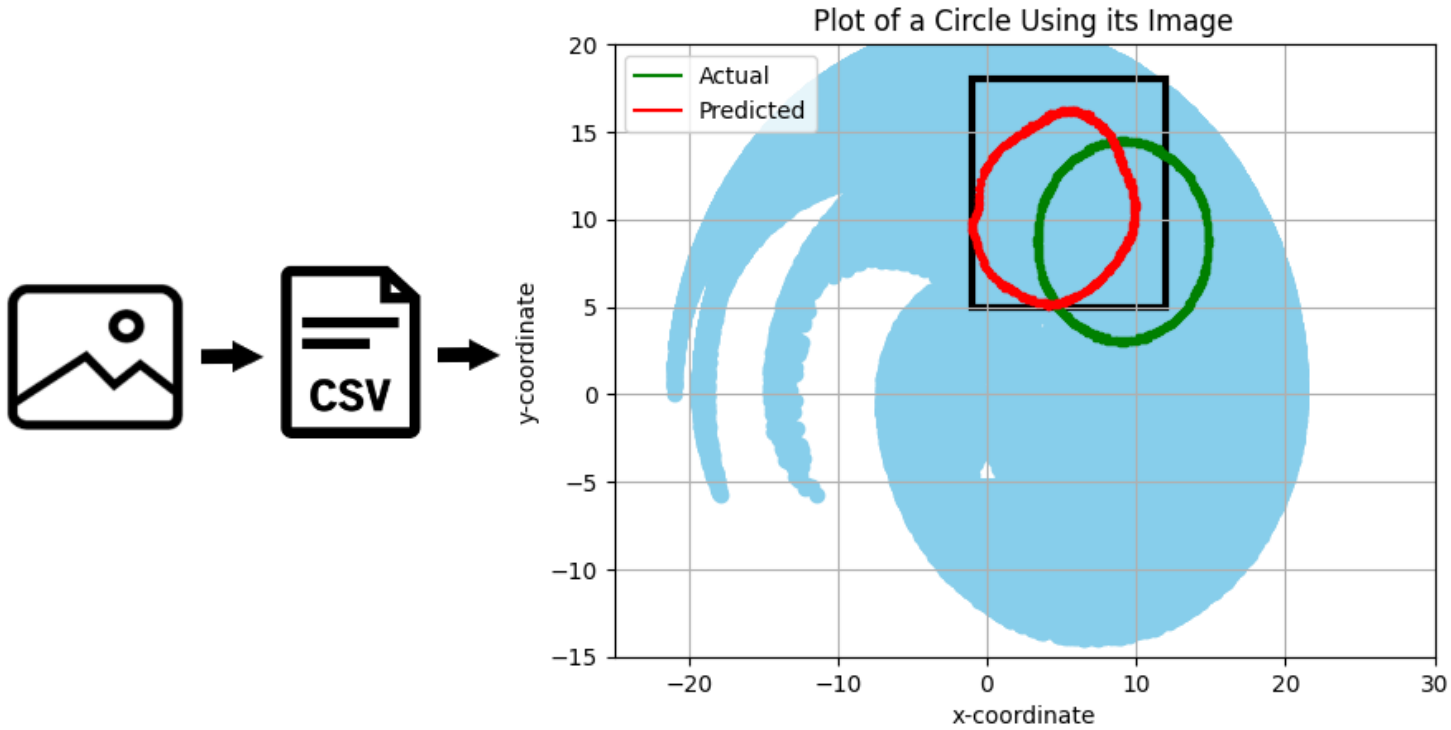

Figure 32: Plot of a Circle

Plot Generated from a Text Input

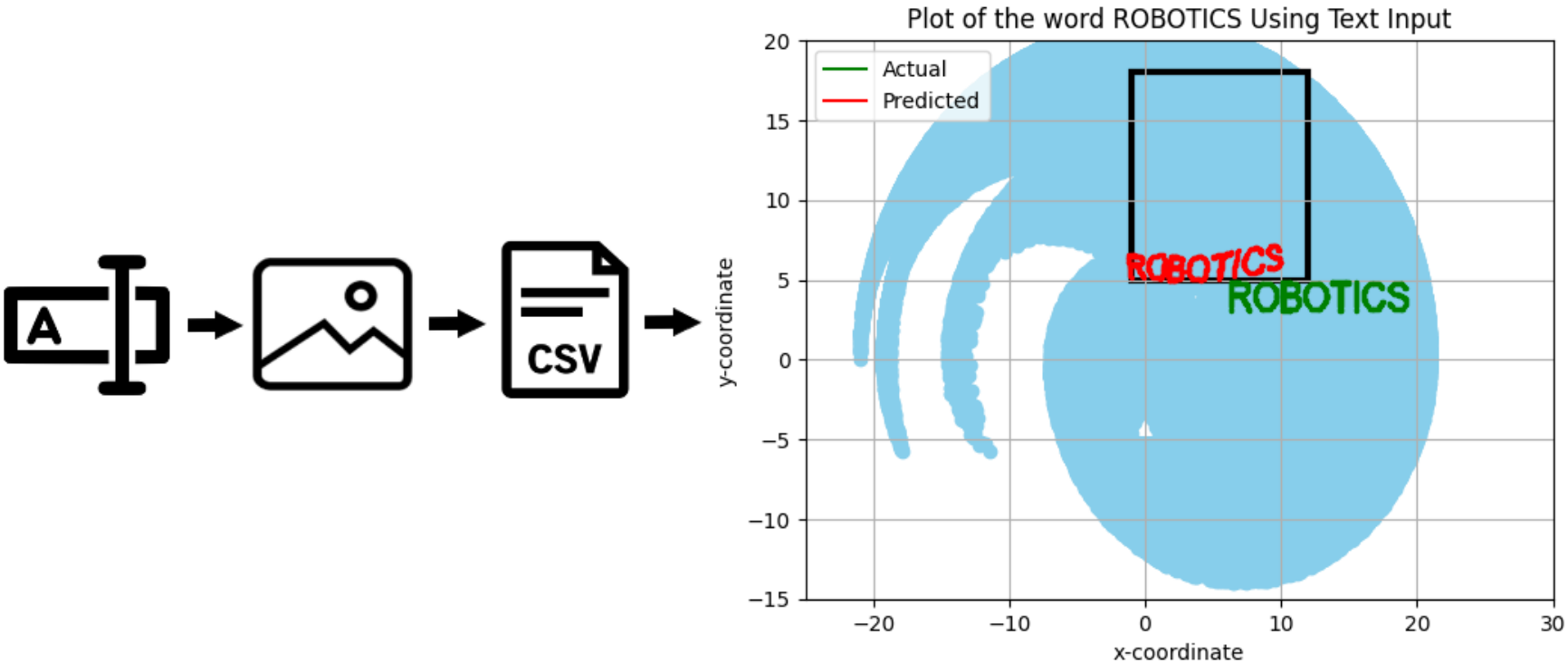

Figure 33: Plot of the word "ROBOTICS"

Plot Generated from a Text Input

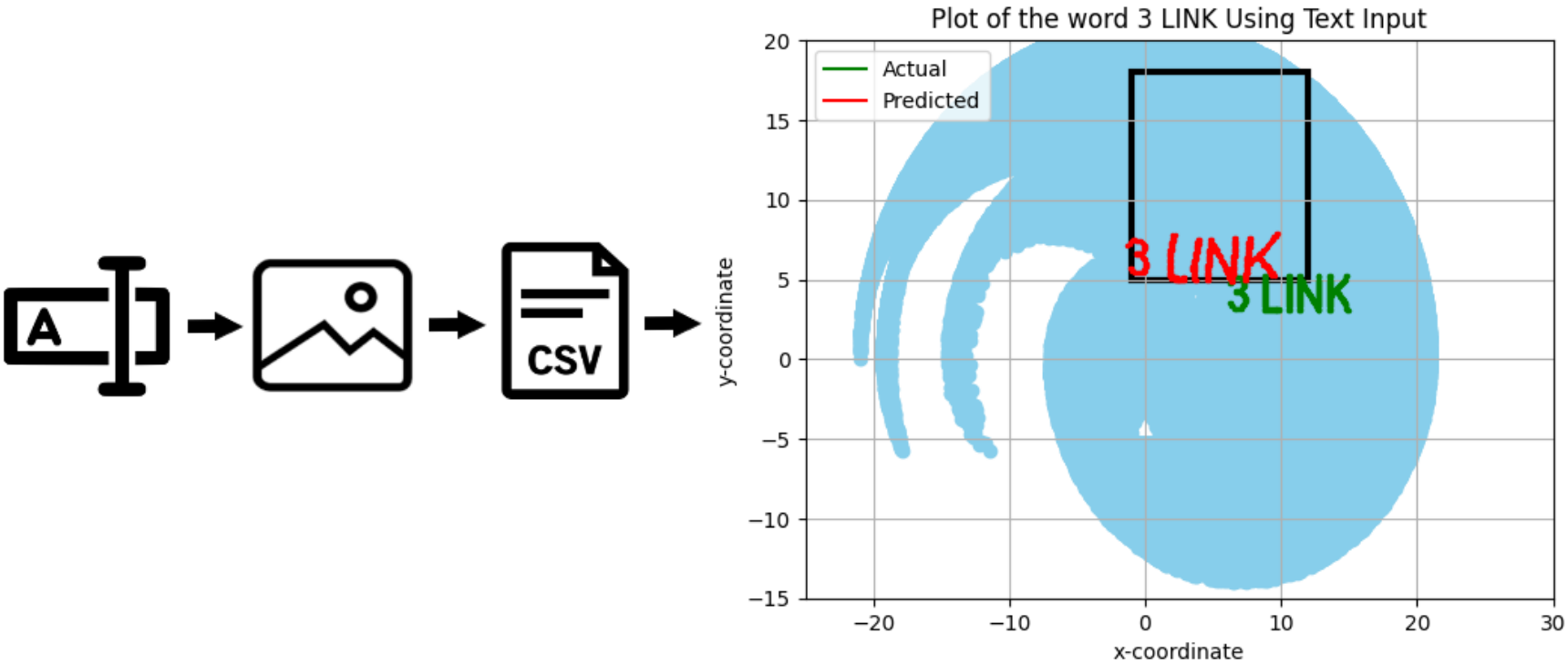

Figure 34: Plot of the word "3 LINK"

Plot Generated from a Text Input

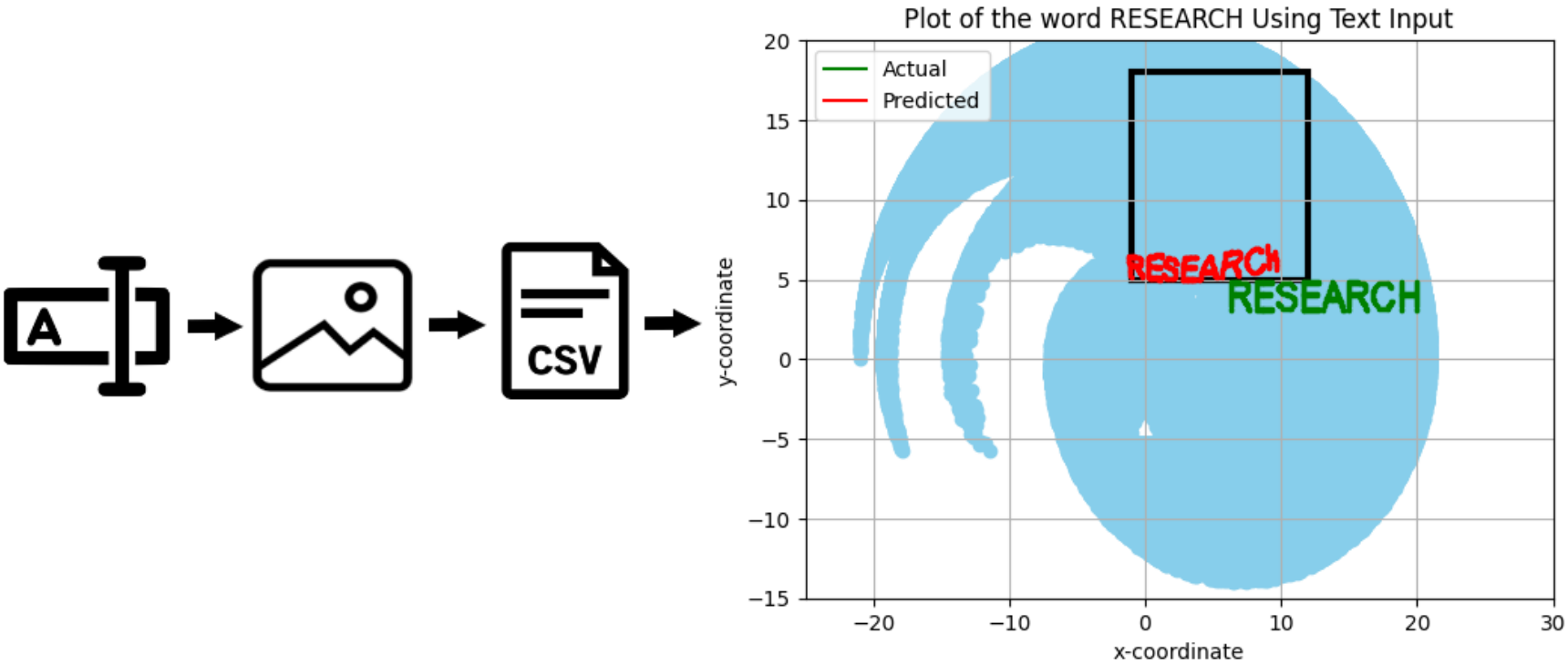

Figure 35: Plot of the word "RESEARCH"

Plot Generated from a Text Input

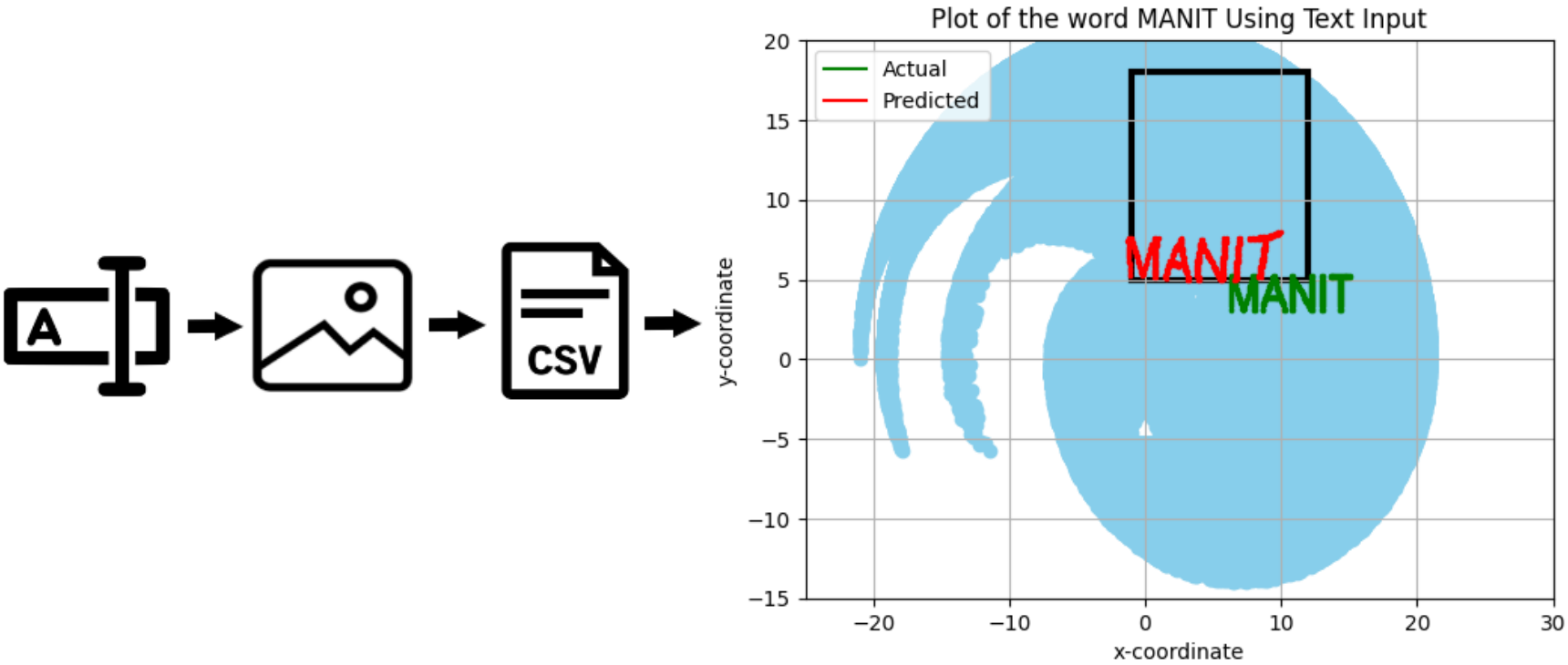

Figure 36: Plot of the word "MANIT"

Plot Generated from a Text Input

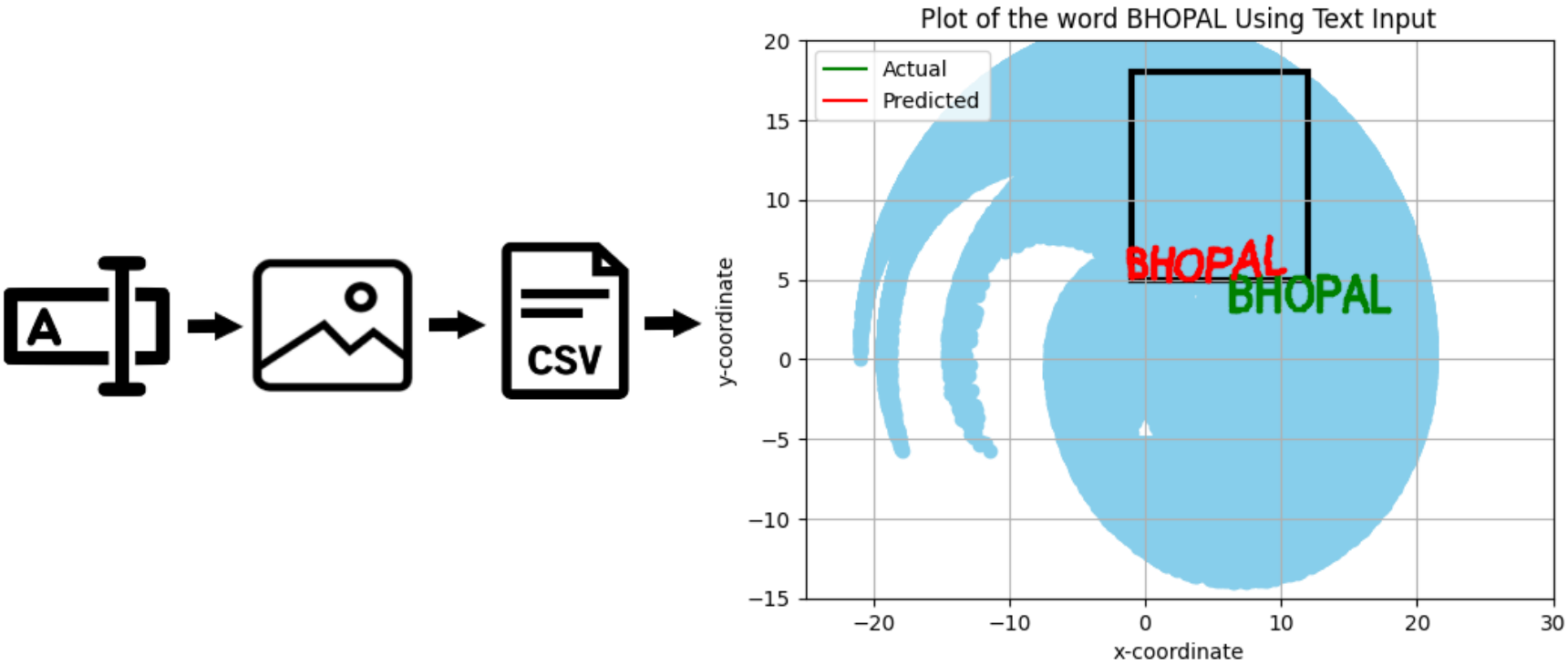

Figure 37: Plot of the word "BHOPAL"

Performance Analysis

Based on the results obtained for x, y, $\theta, \theta_1, \theta_2, \theta_3$ from various experiments, we can evaluate the Model Performance based on metrics as follows.

- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- Mean Absolute Error (MAE)

Performance Analysis for the x - coordinate

The following table shows the various errors for predicting the x-coordinate.

Source	Plot	MSE	RMSE	MAE
CSV	Circle	0.1070	0.3271	0.2410
CSV	MANIT	0.1889	0.4346	0.3152
Image	Circle	0.1003	0.3167	0.2804
Image	Triangle	0.1207	0.3475	0.2367
Image	Square	0.2152	0.4639	0.3281
Image	Pentagon	0.1187	0.3446	0.3004
Image	Hexagon	0.1006	0.3172	0.2762
Text	MANIT	0.0401	0.2001	0.1738
Text	ROBOTICS	0.0673	0.2595	0.2052
Text	3 LINK	0.0365	0.1911	0.1641
Text	BHOPAL	0.0359	0.1897	0.1587
Text	RESEARCH	0.0656	0.2561	0.2024

Performance Analysis for the y - coordinate

The following table shows the various errors for predicting the y-coordinate.

Source	Plot	MSE	RMSE	MAE
CSV	Circle	0.4896	0.6997	0.5170
CSV	MANIT	0.2274	0.4769	0.3704
Image	Circle	1.0671	1.0329	0.6703
Image	Triangle	0.5001	0.7071	0.4301
Image	Square	0.8472	0.9205	0.6214
Image	Pentagon	0.5962	0.7721	0.4574
Image	Hexagon	0.8996	0.9485	0.6020
Text	MANIT	0.0226	0.1504	0.1366
Text	ROBOTICS	0.1807	0.4252	0.2846
Text	3 LINK	0.0232	0.1523	0.1421
Text	BHOPAL	0.0541	0.2324	0.1879
Text	RESEARCH	0.2668	0.5166	0.3264

Performance Analysis for the Angle θ

The following table shows the various errors for predicting the angle θ.

Source	Plot	MSE	RMSE	MAE
CSV	Circle	2.8510	1.6885	1.4429
CSV	MANIT	2.3700	1.5395	1.2901
Image	Circle	2.1737	1.4744	1.2122
Image	Triangle	1.6443	1.2823	1.0863
Image	Square	2.3748	1.5410	1.2571
Image	Pentagon	1.6741	1.2938	1.0524
Image	Hexagon	2.1509	1.4666	1.2085
Text	MANIT	0.6282	0.7926	0.6678
Text	ROBOTICS	1.2983	1.1394	0.8923
Text	3 LINK	0.4295	0.6553	0.5045
Text	BHOPAL	0.7937	0.8908	0.7919
Text	RESEARCH	1.6087	1.2683	0.9833

Performance Analysis for the Angle θ_1

The following table shows the various errors for predicting the angle θ_1.

Source	Plot	MSE	RMSE	MAE
CSV	Circle	2.1010	1.4495	1.2296
CSV	MANIT	1.4034	1.1847	1.0149
Image	Circle	2.4440	1.5633	1.2924
Image	Triangle	1.7767	1.3329	0.9681
Image	Square	1.7676	1.3295	0.9792
Image	Pentagon	2.6667	1.6330	1.3240
Image	Hexagon	2.6074	1.6147	1.3506
Text	MANIT	3.6006	1.8975	1.7710
Text	ROBOTICS	2.8709	1.6943	1.4874
Text	3 LINK	3.1936	1.7871	1.6715
Text	BHOPAL	3.1097	1.7634	1.5335
Text	RESEARCH	2.7132	1.6471	1.4093

Performance Analysis for the Angle θ_2

The following table shows the various errors for predicting the angle θ_2.

Source	Plot	MSE	RMSE	MAE
CSV	Circle	3.8587 e-07	0.0006	0.0005
CSV	MANIT	7.0878 e-07	0.0008	0.0005
Image	Circle	0.0001	0.0104	0.0025
Image	Triangle	0.0057	0.0756	0.0273
Image	Square	0.0074	0.0859	0.0314
Image	Pentagon	0.0018	0.0432	0.0135
Image	Hexagon	3.9301 e-06	0.0019	0.0006
Text	MANIT	2.0139 e-07	0.0004	0.0003
Text	ROBOTICS	2.9717 e-07	0.0005	0.0004
Text	3 LINK	2.0249 e-07	0.0004	0.0004
Text	BHOPAL	2.3576 e-07	0.0005	0.0004
Text	RESEARCH	6.4201 e-07	0.0008	0.0005

Performance Analysis for the Angle θ_3

The following table shows the various errors for predicting the angle θ_3.

Source	Plot	MSE	RMSE	MAE
CSV	Circle	2.1011	1.4495	1.2296
CSV	MANIT	1.4034	1.1846	1.0149
Image	Circle	2.4441	1.5634	1.2939
Image	Triangle	1.7840	1.3357	0.9946
Image	Square	1.7781	1.3334	1.0099
Image	Pentagon	2.6686	1.6336	1.3364
Image	Hexagon	2.6075	1.6147	1.3506
Text	MANIT	3.6006	1.8975	1.7710
Text	ROBOTICS	2.8709	1.6944	1.4874
Text	3 LINK	3.1935	1.7870	1.6714
Text	BHOPAL	3.1098	1.7634	1.5334
Text	RESEARCH	2.7132	1.6472	1.4092

Conclusion

The development of a Neural Network capable of predicting joint angles based on input coordinates is a significant achievement.

The multi-modal nature of the model expands its potential applications across multiple disciplines.

It can be employed in robotics, computer vision, and other fields where the ability to interpret diverse types of information is valuable.

As the model can handle diverse inputs, it aligns with the complex and dynamic nature of robotics applications.

Future Scope

A significant drawback of this model is non-optimal movement of the manipulator.

Due to the non-linear nature of the equations of Forward Kinematics, multiple combinations of joint angles may be possible for a given input.

Thus, there might be multiple ways to transition from one state to the other.

It is important to optimize the transitions such that the new state can be obtained from the previous state with least possible effort.

References

Adrian-Vasile Duka (2014)

Neural Network based Inverse Kinematics Solution for Trajectory Tracking of a Robotic Arm

Elsevier

Henrique Pacini, Kweku Attafuah-Wadee

How to tackle pollution fuelled by manufacturing in developing countries

https://unctad.org/news/blog-how-tackle-pollution-fuelled-manufacturing-developing-countries

Chris Baraniuk

The extreme robot arm that can chop up a ship

https://www.bbc.com/news/business-66881323

José Bettencourt, Juan Guillermo Martín

Underwater Archaeology

https://www.sciencedirect.com/topics/social-sciences/underwater-archaeology

References

Daegil Park, Jong-Boo Han, Teakyeong Yeu, Su-gil Cho, Seongsoon Kim, Hyungwoo Kim, Yeongjun Lee (2023)

Development of an Autonomous Cleaning Robot with a Hydraulic Manipulator Arm for the Cleaning of Niche Areas of a Ship Hull

Journal of Marine Science and Engineering

Jody DeLuca

6 ways Robots can be used to address Unsafe Working Conditions

https://ecorobotics.com/6-ways-robots-can-be-used-to-address-unsafe-working-conditions/

Ewart Cox

Handle with care: robot-assisted system for safe handling of nuclear waste

https://www.hazardexonthenet.net/article/191492/Handle-with-care–robot-assisted-system-for-safe-handling-of-nuclear-waste.aspx

Tai Ha, Jong Il Park, Junsik Park (2019)

Accelerating Deep Learning Based Large-Scale Inverse Kinematics with Intel Distribution of OpenVINO™ Toolkit

https://www.intel.com/content/www/us/en/developer/articles/technical/accelerating-deep-learning-based-large-scale-inverse-kinematics-with-intel-distribution-of.html

References

Ivan Chavdarov, Bozhidar Naydenov (2022)

Algorithm for Determining the Types of Inverse Kinematics Solutions for Sequential Planar Robots and Their Representation in the Configuration Space

MDPI

Wikipedia

Inverse Kinematics

https://en.wikipedia.org/wiki/Inverse$_k$inematics

Nikhil Agnihotri

What are the components of robotic arms and industrial robots?

https://www.engineersgarage.com/robotic-arm-components-construction/

Vijay Bhaskar Semwal, Yash Gupta (Published)

Performance Analysis of Data-Driven Techniques for Solving Inverse Kinematics Problems.

Intelligent Systems Conference (Intellisys), 2021

THE END