Project Directory Overview

Welcome to the **Adobe GenSolve Hackathon (Curvetopia)** repository. This project is focused on developing and testing models and functions related to curve regularization, symmetry detection, and occlusion handling.

The repository is organized into three main sections:

- Sample Test cases
- Models
- Functions

1. `sample_test_cases` Directory

The `sample _test_cases` directory houses the Jupyter Notebooks that demonstrates the model output on the test cases given with the problem statement, used to validate the functionality and performance of our models and functions. The following test cases are included:

- **test case 1, 2, 3 (**which shows Task 1 and Task 2**)**:
    - **Task 1**: **Regularization** of open and closed curves.
    - **Task 2**: Detection of **symmetry**, including **horizontal**, **vertical**, and **diagonal** symmetry.

  Description: This notebook is designed to run through the first set of test scenarios, applying both regularization and symmetry detection algorithms. It tests the model's ability to handle various curve types and assess the accuracy of symmetry identification.

- **test case 4 (**which shows Task 3**)**:
    - **Task 3**: Detection and handling of **occlusions** in curves.

  Description: This notebook focuses exclusively on the task of detecting occlusions within curves. It tests the model's ability to identify and appropriately process occluded sections of a curve, ensuring accurate results even when parts of the curve are missing or obstructed.

2. `Model` Directory

The `Model` directory contains the Jupyter Notebook that encapsulates the models developed for this project. These models are the result of extensive training and optimization processes, each tailored to specific tasks. The models included are:

- **Model 1: Regularizing Open and Closed Curves**

  Purpose: This model is designed to **regularize curves**, ensuring they are smooth and consistent. It works on both open and closed curves, applying different techniques as necessary to achieve optimal results.

- **Model 2: Finding Symmetry**

  Purpose: This model specializes in **detecting symmetry** within curves. It is capable of identifying **horizontal**, **vertical**, and **diagonal** symmetries, providing detailed outputs on the symmetry properties of the input curves.

- **Model 3: Detecting Occlusion Curves**

  Purpose: The occlusion model is tasked with identifying and handling **occlusions** within curves. This involves detecting missing or obstructed sections of the curve and applying techniques to either reconstruct the missing data or process the curve accordingly.

Each model has been trained on a comprehensive dataset, ensuring high accuracy and reliability. The trained models are compiled and stored within this directory for easy access and deployment in various tasks.

3. `Functions` Directory

The `Functions` directory is dedicated to the core functions that underpin the operations within our models and test cases. These functions are implemented in a Jupyter Notebook and are essential for performing the various computational tasks required throughout the project.

Key functions include:

- **split_curve_by_curvature:**

  This function identifies points along a curve where the angle exceeds a specified threshold, allowing the curve to be split at those points. It ensures that segments with sharp turns are treated separately for more accurate processing.

- **dfs_open_curves**:

  This function determines whether a given curve is open, meaning it does not close back on itself. This classification is crucial for applying the correct regularization and symmetry detection techniques.

- **dfs_closed_curves**:

  Similar to the open curve check, this function identifies closed curves, which form a complete loop. Identifying closed curves is essential for applying appropriate regularization methods and ensuring accurate symmetry detection.

- **reconstruct_curve**:

  This function is responsible for reconstructing curves from fragmented or partial data. It plays a critical role in ensuring that our models can handle incomplete data and still produce accurate results.

- **Additional Functions**:

  The notebook includes various other utility functions that assist in the preprocessing, analysis, and manipulation of curve data. These are integral to the overall functionality of the models.