**Project: Artistic Style Transfer Web Application**

**Description**

This project utilizes a pre-trained style transfer model to apply artistic styles to existing images. Users can upload a content image and a style image, and the application will generate a new image where the content image is stylized with the artistic elements of the style image. The application is built using Flask and TensorFlow, with the model hosted on TensorFlow Hub.

**Algorithm**

1. **Input Image Upload**
   * Users upload a content image and a style image through an HTML form.
2. **Image Preprocessing**
   * Resize and normalize the uploaded images to prepare them for the style transfer model.
3. **Style Transfer**
   * Use a pre-trained style transfer model from TensorFlow Hub to apply the style of the style image to the content image.
4. **Image Postprocessing**
   * Convert the resulting tensor image back to an image format and save it.
5. **Output**
   * Display the stylized image to the user and provide a download link.

**Steps**

1. **Initialize Flask Application**
   * Set up directories for uploading images and storing results.
2. **Define Helper Functions**
   * load\_img: Loads and preprocesses images.
   * tensor\_to\_image: Converts a tensor to an image.
3. **Perform Style Transfer Function**
   * perform\_style\_transfer: Applies the style transfer using the pre-trained model.
4. **Define Routes**
   * index: Handles the main page, image upload, and form submission.
   * result: Displays the stylized image.
   * download: Allows the user to download the stylized image.

**Inputs**

1. **Content Image**
   * Uploaded by the user.
2. **Style Image**
   * Uploaded by the user.

**Outputs**

1. **Stylized Image**
   * Generated by the style transfer model and displayed to the user.

**Conditions and Loops**

* **Conditions**
  + Check if both content and style images are uploaded before proceeding with the style transfer.
* **Loops**
  + No explicit loops are used in this script.

**Required Libraries**

* Flask: Web framework for handling requests and rendering templates.
* TensorFlow: Deep learning framework for model loading and image processing.
* TensorFlow Hub: Repository for pre-trained models.
* NumPy: Numerical operations on tensors.
* PIL (Pillow): Image processing library.

**Dataset**

* **Pre-trained Model**
  + https://tfhub.dev/google/magenta/arbitrary-image-stylization-v1-256/2: Pre-trained style transfer model from TensorFlow Hub.