Assignment No X

Submitted to Dr Muhammad Inam ul Haq

Submitted by Muhammad Kamran

Registration No; CS120202034

Department; Computer Science and Bioinformatics

Semester 7th

Subject Computer Vision

Khushal Khan Khattak University, Karak

**Question:** Consider one Image (Base Image) and you have to Compare it with another 20 Images (Testing Images) to Check/Show Similarity Between it, by programming algorithm.

Test\_Images

Test\_Images

**Image Comparison with Structural Similarity Index (SSI)**

**Purpose:**

The code aims to compare a base image against a set of testing images using the Structural Similarity Index (SSI) algorithm and visualize the comparison results.

Library Package Installation in Machine/Computer

* **cv2**: pip install opencv-python
* **numpy**: pip install numpy
* **skimage**: pip install scikit-image
* **matplotlib**: pip install matplotlib

Dependencies:

* **cv2**: OpenCV library for image processing.
* **numpy**: Library for numerical operations.
* **skimage**: Module providing the **structural\_similarity** function for calculating SSIM.
* **matplotlib**: Library for visualization.

Workflow:

1. **Read the Base Image:**
   * The base image is read from the specified file path.
   * If the base image cannot be read, an error message is displayed.
2. **Image Comparison:**
   * The code iterates through a set of testing images, each named sequentially (e.g., Testimage1.jpg, Testimage2.jpg, ...).
   * For each testing image:
     + The image is read from the specified folder path.
     + The testing image is resized to a common size (width and height) for consistency.
     + Both the base image and the testing image are converted to grayscale.
     + The Structural Similarity Index (SSI) is calculated between the base and testing images using the **structural\_similarity** function.
     + The similarity percentage between the images is computed.
     + The base image, testing image, and similarity percentage are stored in lists for visualization.
3. **Visualizing Comparison Results:**
   * The code generates a single window displaying a grid of comparison results using **matplotlib**.
   * Each cell in the grid represents a comparison between a testing image and the base image.
   * The testing images are displayed with their respective similarity percentages as titles.

Usage:

1. **File Paths Setup:**
   * Provide the complete file paths to the base image and the folder containing testing images.
   * Ensure correct file paths and image names according to your directory structure.
2. **Adjust Image Sizes (Optional):**
   * Modify the **width** and **height** variables to resize images to a common size for comparison.
3. **Run the Code:**
   * Execute the Python script in an environment with the required libraries installed (OpenCV, NumPy, scikit-image, and Matplotlib).
4. **Output:**
   * The output is a single window displaying a grid of comparison results.
   * Each cell in the grid contains a testing image alongside its similarity percentage compared to the base image.

Notes:

* Ensure the folder structure, file paths, and image names match the code specifications.
* Verify the compatibility of image formats (e.g., JPG) and ensure proper access rights to read the images.