Report for Group Project: Data Augmentation in Computer Vision

Part B: Practical Implementation of Data Augmentation Techniques

Q3: Data Augmentation using PyTorch

Steps and Implementation

1. **Importing Libraries:** We started by importing necessary libraries, including `os` for directory operations, `PIL` for image processing, `torchvision.transforms` for applying transformations, and metrics from `skimage` for image comparison.
2. **Reading Images:** We loaded the images from the specified directory and stored them in a list.
3. **Applying Transformations:** We defined a set of transformations using `transforms.Compose()`. These transformations included random affine transformations, horizontal flips, rotations, and color jittering.
4. **Generating Augmented Images:** For each image, we generated five augmented versions using the defined transformations. These images were saved in a new directory.
5. **Comparing Images:** We compared the original and augmented images using SSIM and MSE metrics. The results were stored in a DataFrame for analysis.

Results and Analysis

- **SSIM Results:** The SSIM values were consistently below 0.5, indicating significant differences between the original and augmented images.
- **MSE Results:** The MSE values were high, further confirming the substantial differences.

These results suggest that the transformations introduced significant variations in the images, effectively augmenting the dataset.

Q4: Photometric Distortions using OpenCV

Steps and Implementation

1. **Importing Libraries:** Similar to Q3, we imported necessary libraries for image processing and comparison.
2. **Defining Photometric Functions:** We implemented functions to adjust brightness, contrast, and saturation, as well as shift color channels.
3. **Applying Transformations:** We applied these functions to images and saved the results.
4. **Comparing Images:** We compared the original and augmented images using SSIM and MSE metrics, storing the results in a DataFrame.
5. **Visualizing Results:** We used `matplotlib` to visualize the original and augmented images for each photometric adjustment.

Results and Analysis

- **SSIM Results:** The SSIM values were close to 1 for most transformations, indicating high similarity with the original images.
- **MSE Results:** The MSE values were low, further confirming the high similarity.

These results suggest that the implemented photometric transformations were effective in maintaining the overall appearance of the images while introducing necessary variations.

Use of Generative AI

Generative AI was used to enhance this documentation based on the provided notes. Specifically, ChatGPT assisted in:

- Refining the explanation of photometric distortion techniques.
- Suggesting effective methods for comparing augmented images.
- Providing guidance on structuring the report.

The use of generative AI has been declared in the Generative AI Journal and followed the project's ethical guidelines.