USN:23BTRCAO55
NAME: G CHANIKYA

Industrial AI Activity

LIBRARY: MedPy

Overview:

MedPy is a Python library designed specifically for medical image processing.

It provides tools for handling and analyzing 2D and 3D medical images such as CT, MRI, and ultrasound scans.

It supports common medical image formats like DICOM, NIfTI, and Analyze, and integrates seamlessly with NumPy and SciPy.

Typical use cases include: - Image filtering and enhancement - Image segmentation - Feature extraction - Morphological operations - Distance and similarity measurements

Functions in MedPy Library:

Function / Module	Description
medpy.io.load()	Loads medical image files (DICOM, NIfTI, etc.) into arrays.
medpy.io.save()	Saves NumPy arrays back into medical image formats.
medpy.filter.smoothing.anisotropic_diffusion()	Reduces noise in images while preserving edges.
medpy.filter.smoothing.gaussian()	Applies Gaussian smoothing to images.
medpy.metric.binary.dc()	Computes the Dice Coefficient – a measure of image similarity.
medpy.metric.binary.hd()	Computes the Hausdorff Distance between two segmentations.
medpy.features.intensity.mean_intensity()	Calculates mean pixel intensity.
medpy.filter.binary.dilation()	Performs binary image dilation (increases object size).

Code and Output:

```
# Install MedPy before running:
# pip install medpy

from medpy.io import load, save
from medpy.filter.smoothing import
anisotropic_diffusion from medpy.metric.binary
import dc
import numpy as np

# Load medical image (example image path)
image_data, image_header =
load('brain_scan.nii')

# Apply anisotropic diffusion filtering
filtered_image = anisotropic_diffusion(image_data, niter=10, kappa=50,
gamma=0.1)

# Compare similarity between original and filtered image using Dice
Coefficient # (For demonstration, we threshold both images)
original_binary = (image_data > np.mean(image_data)).astype(np.uint8)
filtered_binary = (filtered_image >
np.mean(filtered_image)).astype(np.uint8)
```

```
dice_score = dc(original_binary, filtered_binary)
print("Original Image Shape:", image_data.shape)
print("Filtered Image Shape:", filtered_image.shape)
print("Dice Coefficient between Original and Filtered:", round(dice_score, 3))
```

Sample Output:

```
Original Image Shape: (256, 256)
Filtered Image Shape: (256, 256)
Dice Coefficient between Original and Filtered: 0.89
```

Conclusion:

MedPy provides a comprehensive set of functions for medical image analysis, including filtering, segmentation, and quantitative comparison of medical scans. It is widely used in AI-driven healthcare research, enabling the development of intelligent diagnostic systems for improved patient care.