Report on Feature Extraction and Similarity Analysis from CT Knee Regions

Overview:

This task involved converting a 2D pre-trained DenseNet121 into a 3D CNN for volumetric knee CT analysis, extracting features from tibia, femur, and background regions using segmentation masks, and computing cosine similarities between regions at multiple convolutional depths. Results were saved in a CSV format.

Task 1 – Segmentation Based Splitting

Goal: Segment femur, tibia and background as regions of interest bones from 3D CT images.

Approach:

- Applied intensity thresholding on CT volume (HU > 320) to isolate bone regions.
- Preprocessed the binary mask by removing small noise components, filling internal holes slice-by-slice, and disconnecting femur and tibia by zeroing the connecting axial slice.
- Identified the two largest connected components using connected component labeling and size filtering.
- Labeled components as femur or tibia based on their spatial position (center of mass along axial axis).
- Used mask labels to isolate region-specific CT volumes where Tibia = 2, Femur = 1 and Background = 0

Task 2 – Convert 2D Pretrained Model to 3D

Goal: Adapt a 2D pretrained DenseNet121 model to process 3D volumes.

Approach:

- Imported DenseNet121 from torchvision.models.
- Replaced all Conv2d layers with Conv3d layers:
    - Inflated kernels by repeating weights along the depth dimension.
    - Divided weights by depth to maintain consistent scaling.
    - Replaced supporting layers:
- BatchNorm2d → BatchNorm3d
    - MaxPool2d(3,2,1) → MaxPool3d((1,3,3), (1,2,2), (0,1,1))
    - AvgPool2d(2,2) → AvgPool3d((1,2,2), (1,2,2))
    - AdaptiveAvgPool2d((1,1)) → AdaptiveAvgPool3d((1,1,1))
- Preserved the behavior of pretrained spatial filters while extending to depth.

Outcome: Outcome: Fully functional 3D DenseNet121 model capable of accepting (B, 3, D, H, W) inputs with minimal deviation from the pretrained model's learned structure.

Task 3 – Feature Extraction

Goal: Extract fixed-length deep feature vectors from multiple depths of the CNN.

Approach:

- Passed each of the 3 region volumes through the inflated 3D DenseNet.
- Captured feature maps from:
    - Last Conv3d layer
    - Third-last Conv3d layer
    - Fifth-last Conv3d layer
- Applied global average pooling (GAP) on each to generate N-dimensional vectors.
- Ensured input shape was (1, 3, D, H, W) by repeating grayscale input along the channel dimension.

Outcome: Obtained 3 pooled feature vectors for each region, used for pairwise comparison.

Task 4 – Feature Comparison:

Goal: Quantify feature similarity between anatomical regions using extracted feature vectors.

Approach:

- For each pair: Tibia ↔ Femur, Tibia ↔ Background, Femur ↔ Background
- Computed cosine similarity between feature vectors from:
    - Last conv layer
    - 3rd-last conv layer
    - 5th-last conv layer
- Used PyTorch's F.cosine_similarity() for comparison.

Outcome: Generated 9 similarity scores per scan, reflecting multi-scale anatomical similarity between regions.

Task 5 – Result Organization:

Goal: Save similarity metrics in a reproducible format in a csv file format

Approach:

- Created CSV file with rows for each pairwise comparison and columns for each feature depth.
- Followed format:
    - Columns: layer_1_sim, layer_3_sim, layer_5_sim
    - Rows: tibia_femur, tibia_background, femur_background
- Uploaded final source code to GitHub, zipped repo for submission.

Outcome: Submission includes a reproducible pipeline, organized outputs, and a .csv file with region similarity scores.