==========

Detailed Guide for Using and Understanding the OsteoCancerNet Pipeline

Author: Nashaat M. Hussein

Date: 2026-01-07

==========

This document provides a detailed explanation of the OsteoCancerNet pipeline, including its purpose, programming language, functionality, and step-by-step instructions for usage. The goal is to enable researchers or practitioners to understand what the code does, how it works, and how to execute it to reproduce the results.

1. Programming Language

The OsteoCancerNet code is written in Python, a widely used language for machine learning and deep learning. It utilizes libraries such as TensorFlow/Keras for neural network operations, OpenCV for image processing, NumPy for numerical computations, Matplotlib for visualization, and scikit-learn for classical machine learning (SVM classifier).

2. Overview of the Pipeline

OsteoCancerNet is a complete end-to-end system for classifying bone X-ray images into two categories: Normal and BoneCancer. The pipeline combines image preprocessing, feature extraction using a pre-trained EfficientNet-B4 model, classification using an SVM, performance evaluation, and Grad-CAM visualization for interpretability. The main steps include:

Image Preprocessing: Enhancing X-ray images using grayscale conversion, CLAHE (contrast enhancement), and Unsharp Masking.

Feature Extraction: Using EfficientNet-B4 (pre-trained on ImageNet) to extract meaningful features from images.

SVM Classification: Training a Support Vector Machine on the extracted features to classify the images.

Data Augmentation: Applying transformations such as rotation, flipping, and brightness adjustments to improve generalization.

Evaluation Metrics: Computing accuracy, precision, recall, F1-score, confusion matrix, and ROC curve.

Grad-CAM Visualization: Highlighting important regions in the images that influence the SVM's predictions.

3. Step-by-Step Usage Instructions

Step 1: Set up the Python environment

Install Python 3.9 or higher.

Create a virtual environment and activate it.

Install required libraries:

pip install tensorflow opencv-python numpy matplotlib scikit-learn

Step 2: Prepare the dataset

Organize images into the following folder structure:

dataset/

train/Normal

train/BoneCancer

val/Normal

val/BoneCancer

test/Normal

test/BoneCancer

Ensure images are in .jpg or .png format.

Step 3: Preprocess the images

Each image is read, resized to 380x380, converted to grayscale.

CLAHE enhances local contrast, Unsharp Masking increases edges.

Images are converted back to RGB for compatibility with EfficientNet.

Step 4: Data augmentation (training set only)

Rotations, horizontal flips, and brightness adjustments are applied to reduce overfitting.

Step 5: Feature extraction

EfficientNet-B4 extracts high-level features from each image.

Preprocessing specific to EfficientNet is applied before feature extraction.

Step 6: Feature standardization

Features are scaled using StandardScaler to zero mean and unit variance.

Step 7: Train the SVM classifier

A radial basis function SVM is trained on the scaled features.

Hyperparameters: C=10, gamma=0.01, kernel=rbf.

Step 8: Predictions and evaluation

The trained SVM predicts classes for train, validation, and test sets.

Accuracy, precision, recall, and F1-score are computed.

Confusion matrix and ROC curve are plotted for performance analysis.

Step 9: Grad-CAM visualization

Grad-CAM is applied to visualize which areas of the X-ray influence predictions.

Heatmaps are overlaid on original images for interpretation.

Step 10: Running the code

Save the provided Python script (e.g., osteocancernet.py).

Ensure dataset paths are correct.

Run the script using:

python osteocancernet.py

Outputs include metrics, plots, and Grad-CAM images.

4. Tips for Reproducibility

Set random seeds for Python, NumPy, and TensorFlow to obtain consistent results.

Maintain consistent preprocessing, augmentation, and SVM parameters.

Use the same train/validation/test split.

For large datasets, use a small subset first to verify functionality.

Conclusion

This guide explains both what the code does and how to use it step by step. By following this document, any researcher can run OsteoCancerNet, understand its workflow, and reproduce classification results on bone X-ray images accurately.