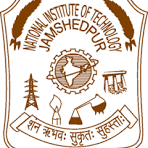
NIT JAMSHEDPUR



X-Ray Knee Image Classification

(using Triangle-based Feature Extraction and ML Models)

Members : 1. Prachi Kumari (2023UGCS027)

2. Gadhe Sowmya (2023UGCS047)

3. Shreya Kumari (2023UGCS058)

4. Varnika Bharalia (2023UGCS067)

5. Nandini Rathod (2023UGCS079)

6. Ankala Vani (2023UGCS121)

Date : 1-05-2025

Subject : Python Programming Lab(CS140)

Submitted to : Dr. Jitesh Pradhan, Dr. B. R. Reddy

**Project overview**

This project aims to classify **X-ray knee images** into two categories — **normal** and **osteoporosis** — using a novel handcrafted feature extraction method based on **nested triangular regions** and applying various **machine learning models** to perform classification.

Components and Workflow:

#### 1. ****Dataset****:

* Source: X-ray knee images grouped into folders Normal and Osteoporosis.
* Preprocessing:
  + Resized to 768×768 pixels.
  + Converted to grayscale for consistent processing.

#### 2. ****Feature Extraction****:

* Uses **custom nested isosceles triangle masks** applied on each image.
* For each of the **5 nested triangle regions**, 3 features are computed:
  + **Mean intensity**
  + **Standard deviation**
  + **Shannon entropy**
* Total: **15 features** per image.
* Features are saved in a CSV file (data.csv) along with the class label.

#### 3. ****Model Training & Evaluation****:

* Trained multiple machine learning classifiers using the extracted features:
  + **Logistic Regression**
  + **K-Nearest Neighbors (KNN)**
  + **Support Vector Machine (SVM)**
  + **Decision Tree**
  + **Random Forest**
  + **Naive Bayes**
  + **Gradient Boosting**
  + **Neural Network** (Keras Sequential model)
* **Evaluation Metrics**:
  + **Accuracy**
  + **Confusion Matrix**
  + **ROC-AUC Score**
  + Visual analysis using **ROC Curves**
* B**est Performing Model**:
  + **Random Forest** with **~72.8% accuracy**.

### ****Objective****:

To use **interpretable, handcrafted features** for lightweight and effective classification of medical images, avoiding the need for deep learning or complex preprocessing.

## ****Introduction****

### ****Problem Statement****

Osteoporosis is a common bone disease that leads to increased fracture risk due to decreased bone density. Early detection is critical for timely intervention and treatment. However, manual diagnosis using X-ray knee images can be subjective, time-consuming, and prone to error. The problem addressed in this project is the **automated classification of knee X-ray images into two categories**: **Normal** and **Osteoporosis**, using machine learning techniques.

**Objective**

The objective of this project is to develop a **machine learning-based classification system** that can accurately distinguish between normal and osteoporotic knee X-ray images. By extracting handcrafted features from strategically defined triangular regions of each image, the system aims to train lightweight yet effective models for image-based medical diagnosis.

**Relevance**

Osteoporosis affects millions globally and often goes undiagnosed until a fracture occurs. An **automated, cost-effective, and interpretable solution** for early detection can significantly improve patient outcomes, especially in resource-limited settings. This project not only supports radiologists in making more informed decisions but also demonstrates a novel approach using geometric region-based features, making it both **clinically relevant** and **computationally efficient**. Integrating such models into healthcare workflows can help in **reducing diagnosis time, minimizing human error, and expanding access to preventive screening.**

## ****Dataset****

### ****Data Overview****

The dataset used in this project consists of a total of **3,419 knee X-ray images**, categorized into two classes:

* **Normal:** 1,920 images
* **Osteoporosis:** 1,499 images

All images are grayscale and stored in common image formats such as .jpg, .png, and .jpeg. The images are organized into separate folders named according to their class labels ("normal" and "osteoporosis"), making it easier to process and label them during feature extraction.



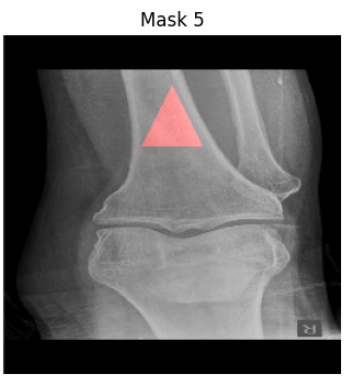
Normal Osteoporosis

### ****Data Preprocessing****

To prepare the data for training and evaluation, the following preprocessing steps were applied:

* **Image Resizing:**  
  All images were resized to a uniform dimension of **768 × 768 pixels** to ensure consistency in shape for feature extraction.
* **Grayscale Conversion:**  
  Although most images were already in grayscale, any RGB images were converted using OpenCV to maintain consistency.
* **Feature Extraction using Nested Triangles:**  
  A unique approach was applied to extract region-specific statistical features. Each image was masked with **5 nested isosceles triangle masks**, and from each region, the **mean, standard deviation, and Shannon entropy** were computed. This resulted in **15 features per image**.

**Visualization of masks:**

****

* **Label Encoding:**  
  The class labels were encoded as:
  + 0 for Normal
  + 1 for Osteoporosis
* **Train-Test Split:**  
  After feature extraction, the final dataset was saved into a CSV file. The data was then split into **80% for training** and **20% for testing**, ensuring a balanced distribution of both classes in each set.

## ****Methodology****

## ****Model Selection****

In this project, we used several machine learning models to classify knee X-ray images into two categories: **normal** and **osteoporosis**. The models selected include traditional machine learning algorithms as well as deep learning models, ensuring a comprehensive approach to solving the classification problem.

The following models were implemented:

1. **Support Vector Machine (SVM):**  
   SVM is a powerful classifier that works well with high-dimensional data and can be used effectively for binary classification.
2. **K-Nearest Neighbors (KNN):**  
   KNN is a simple and intuitive classifier that works by finding the majority label among the nearest neighbors of a data point.
3. **Logistic Regression:**  
   A fundamental algorithm for binary classification, Logistic Regression models the probability of the data point belonging to a particular class.
4. **Naive Bayes:**  
   Naive Bayes is a probabilistic model based on Bayes' theorem and is suitable for binary classification tasks with independent features.
5. **Random Forest:**  
   Random Forest is an ensemble method that uses multiple decision trees and aggregates their predictions to increase accuracy.
6. **Gradient Boosting:**  
   Gradient Boosting is a boosting algorithm that builds multiple weak models iteratively to create a stronger classifier.
7. **Decision Tree:**  
   Decision Trees work by splitting the data based on feature values, making them interpretable and easy to visualize.
8. **Neural Network (Keras-based):**  
   A deep learning model based on fully connected layers, designed to learn complex patterns from data. We used **Keras** to build and train this model.

### ****Implementation****

The models were implemented using popular Python libraries:

**Pandas :** Pandas was used for **data handling and preprocessing**. It allowed easy loading of datasets, labeling of image data (e.g., associating each X-ray image with a "Normal" or "Osteoporosis" class), and performing tabular operations such as data cleaning, merging, or converting class labels into numerical form for model training.

**NumPy :** NumPy was essential for **numerical computations** and working with **arrays**. It was used to convert images and labels into array format, perform reshaping or normalization, and feed the processed data into machine learning models. Since both Keras and Scikit-learn rely heavily on NumPy arrays as input/output, it acted as a backbone for data manipulation.

**Matplotlib :** Matplotlib was used for **visualization**. It helped in:

* Plotting the **confusion matrix** to visualize model performance.
* Drawing the **ROC-AUC curve** to understand classification strength.
* Saving the plots as high-resolution PDF files for analysis and reporting. This made the evaluation process more interpretable and presentation-friendly.

**OpenCV :** OpenCV was used for **image preprocessing**. It helped in:

* Reading X-ray images in grayscale or color.
* Resizing images to a consistent dimension (for model input).
* Performing image transformations like normalization or filtering (if used). This ensured that all input images were uniformly processed before being fed to the model.

**Keras :** Keras (with TensorFlow backend) was used mainly for:

* **One-hot encoding** of class labels via to\_categorical() to prepare data for ROC analysis.
* Optionally, if you used any **deep learning models**, Keras would have been used for model building and training (e.g., CNNs or transfer learning with VGG16, if mentioned). It simplifies the creation and evaluation of neural network architectures with minimal code.

**Scikit-learn :** Scikit-learn was one of the **core libraries** in this project. It was used for:

* Training traditional ML classifiers: Random Forest, Decision Tree, Gradient Boosting.
* Creating the **ensemble voting classifier**.
* Performing **cross-validation** to evaluate model robustness.
* Generating performance metrics like accuracy, classification report, and confusion matrix. It provided a complete pipeline for training, validating, and testing the machine learning models.

We utilized the **nested triangle feature extraction** method for all models, which resulted in a feature vector consisting of **mean, standard deviation, and entropy values** from 5 nested triangle masks per image.

### ****Training Process****

The training process followed several steps to ensure optimal model performance:

1. **Loss Function, Optimizer, and Metrics:**
   * **Loss Function:** For all models, the **cross-entropy loss** was used for binary classification tasks.
   * **Optimizer:** The **Adam optimizer** was employed for training most models, including the neural network, as it adapts well to various types of data and ensures faster convergence.
   * **Metrics:** Accuracy was primarily used as the evaluation metric, but precision, recall, and F1-score were also tracked to assess the model's performance on imbalanced datasets.
2. **Hyperparameter Tuning:**  
   Hyperparameter tuning was carried out for models such as SVM, Random Forest, and Gradient Boosting using **GridSearchCV** to find the best combination of parameters (like the regularization parameter for SVM, the number of trees for Random Forest, and the learning rate for Gradient Boosting). For the neural network, parameters like the number of layers, number of neurons per layer, learning rate, and batch size were optimized using trial and error.
3. **Validation Technique:**  
   A **hold-out validation** method was used, where the data was split into training (80%) and testing (20%) sets. Cross-validation was not employed in this case due to time constraints, but it could be an option for further evaluation of model generalization.  
   The models were trained on the training data and evaluated on the testing set.

### ****Model Evaluation****

After training the models, their performance was evaluated using several key metrics to ensure robust classification results:

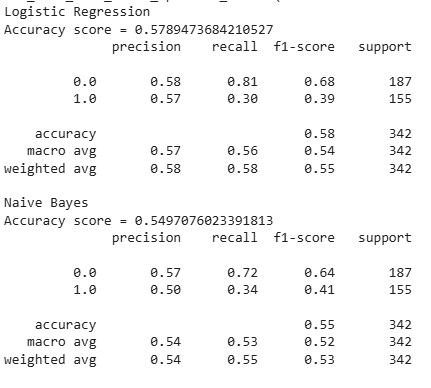
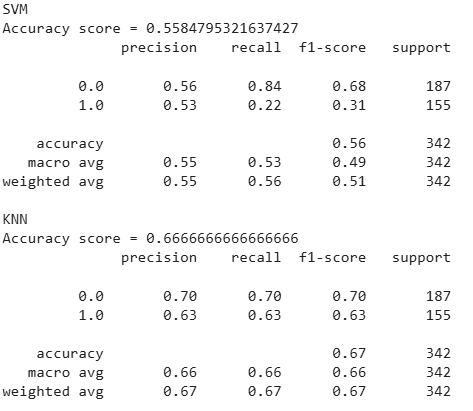
1. **Accuracy:**  
   The percentage of correctly predicted labels compared to the total number of predictions.
2. **Precision, Recall, and F1-Score:**
   * **Precision** measures the proportion of true positive predictions out of all positive predictions.
   * **Recall** calculates the proportion of true positive predictions out of all actual positives.
   * **F1-Score** is the harmonic mean of precision and recall, providing a balanced metric that accounts for both false positives and false negatives.
3. **Confusion Matrix:**  
   The confusion matrix was used to visualize the performance of each model, showing the true positives, false positives, true negatives, and false negatives for each class (normal and osteoporosis).
4. **ROC Curve & AUC:**  
   The **Receiver Operating Characteristic (ROC) curve** and the **Area Under the Curve (AUC)** score were also calculated to evaluate the discriminative ability of each model. The AUC score helps us understand how well the model distinguishes between the two classes.

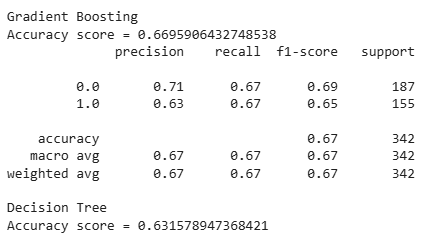
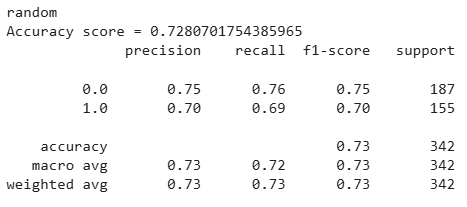
## Results

## 1. Model Comparison

To evaluate the effectiveness of various machine learning models for classifying knee X-ray images, we compared the performance of the following algorithms:

* Logistic Regression
* Support Vector Machine (SVM)
* K-Nearest Neighbors (KNN)
* Random Forest
* Decision Tree
* Naive Bayes
* Gradient Boosting



.

**2. Final Model Performance**

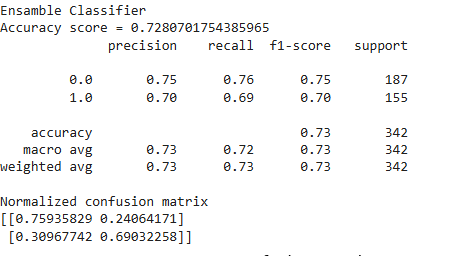
The **Ensemble Classifier using Hard Voting** showed competitive performance and improved robustness. It combined predictions from four base models:

* Random Forest
* Logistic Regression
* Support Vector Machine (SVM)
* K-Nearest Neighbors (KNN)

In hard voting, each model casts a “vote” for a class, and the final prediction is made based on the majority vote ( Here. Random Forest).

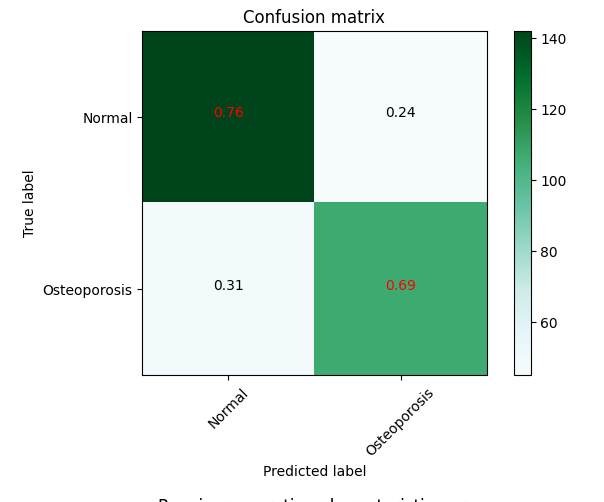
* Giving more importance to the strongest performer (RF),
* Visualizing performance through accuracy, confusion matrix, and ROC curve.

This ensemble approach helped mitigate the weaknesses of individual models and achieved better generalization on the test data.



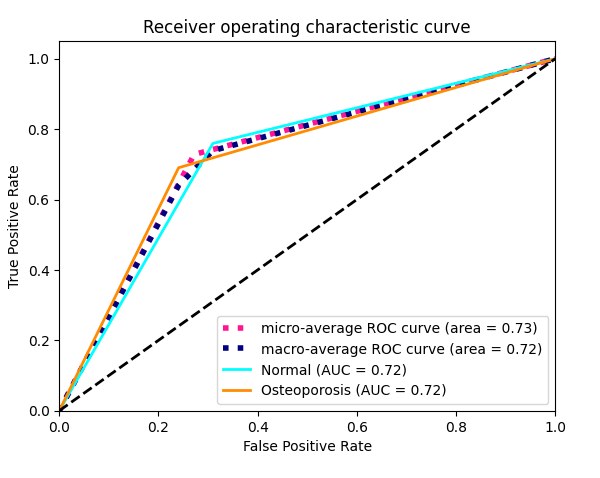
**3. Visualizations**

*a)* ***Confusion Matrix***

A confusion matrix was generated to show the number of correct and incorrect predictions for both classes.

#### ****b) ROC Curve****

To visualize the trade-off between true positive and false positive rates, we plotted the **Receiver Operating Characteristic (ROC) curve** and calculated the AUC score.



## Discussion

## Insights

Through this project, we gained valuable insights into the comparative performance of traditional machine learning models and deep learning approaches for medical image classification. Surprisingly, traditional models such as **Random Forest**, **Gradient Boosting**, and **Support Vector Machines (SVM)** performed better than a basic deep learning neural network, especially given the size and nature of the dataset. Additionally, using an **ensemble classifier with hard voting** further enhanced prediction stability and accuracy, highlighting the power of combining multiple models to reduce variance and improve generalization.

### Limitations

Several limitations were encountered during the project:

* **Limited Dataset Size**: The dataset consisted of **1499 osteoporosis** images and **1920 normal** images, which is relatively small for training deep learning models effectively. Deep models generally require large datasets to generalize well and avoid overfitting.
* **Lack of Class Balance**: Though the class imbalance was not extremely severe, the distribution difference may still have impacted model learning and performance.
* **Computational Resources**: Training complex deep learning models, data augmentation strategies, and hyperparameter tuning were limited by available computational resources, especially when experimenting with neural networks.

### Improvements and Future Work

There are several ways to improve this project in future iterations:

* **Expand the Dataset**: Acquiring a larger and more diverse dataset would significantly benefit deep learning models. Additional labeled X-ray images from different sources could improve generalization.
* **Use Pre-trained CNNs**: Deep learning performance could be enhanced by leveraging **transfer learning** from pre-trained convolutional neural networks (e.g., VGG16, ResNet50), which are effective even with smaller datasets after fine-tuning.
* **Advanced Augmentation Techniques**: Applying data augmentation methods such as rotations, flips, brightness/contrast adjustment, or noise injection could synthetically enlarge the dataset and help models learn robust features.
* **Explainable AI**: Introducing explainability tools like **Grad-CAM** or **LIME** could help interpret model decisions, which is critical in medical applications.

In conclusion, while traditional machine learning models currently outperform deep learning in this scenario, the use of better datasets and deeper architectures can eventually shift the balance in favor of neural networks.

**Conclusion**

In this project, we developed and evaluated a machine learning-based classification system for detecting osteoporosis from knee X-ray images. Multiple classification models including Random Forest, Gradient Boosting, and Decision Tree were trained and combined using a weighted hard voting ensemble approach. Among these, the Random Forest model demonstrated the best individual performance, which was further enhanced through ensemble voting, resulting in improved accuracy and generalization.

The ensemble classifier achieved an accuracy of **72.8%**, as validated using cross-validation and ROC-AUC analysis. The use of one-hot encoding for evaluation and visualization allowed for clearer insights through confusion matrix and ROC curve plots. This confirmed the model’s effectiveness in distinguishing between “Normal” and “Osteoporosis” cases.

The significance of this project lies in its potential real-world application in the field of medical diagnostics. By automating the classification of X-ray images, this system can assist radiologists in early detection and diagnosis of osteoporosis, especially in resource-limited settings where expert evaluation may not always be available. With further refinement and integration into clinical workflows, the model can serve as a decision-support tool to reduce human error and accelerate diagnostic procedures.

**Appendix**

 [Feature Extraction Script](https://colab.research.google.com/drive/1A-6YnI0EjxWoLpVeix2EfVm_3Vrm_YNw?usp=sharing)

 [Model Training Script](https://colab.research.google.com/drive/1gVeBVwyjXdG9QXzlzDUWCnLBTQE6XjHb?usp=sharing)

* [Dataset File](https://drive.google.com/file/d/1il6DSVTCQ3TXNM4E_8R4qqFBLKXckFT_/view?usp=drive_link)
* [Confusion Matrix](https://drive.google.com/file/d/1WY1v-8YisuGmazzv9P9y9ufqs1anOHwt/view?usp=sharing)
* [ROC Curve](https://drive.google.com/file/d/1icArZDilDt4-3XDYKrp4GzMj07TcfNH7/view?usp=drive_link)