Knee Osteoarthritis Prediction using Deep Learning

By Sri Sankeerth Koduru, Ali Irtaza and Anantha

Introduction

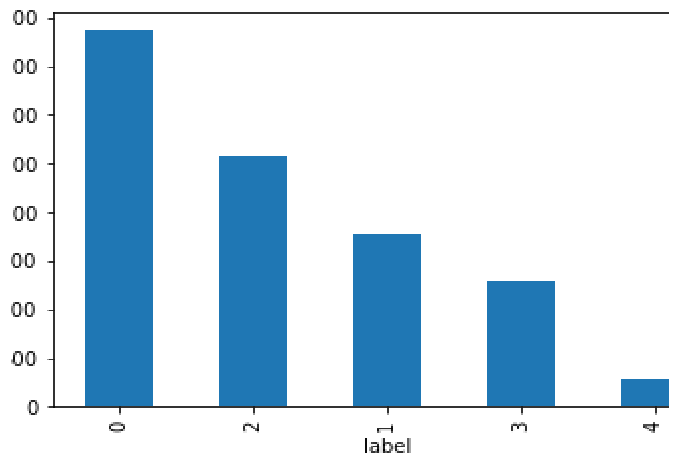
- This project aims to develop an automated system for classifying the severity of knee osteoarthritis (OA) using X-ray images.
- Utilizing advanced deep learning models (VGG19 and Xception) with transfer learning and customized weights.
- Implementing a 5-class classification scheme for severity assessment.
- Employing custom image processing techniques for optimal model performance.

Data

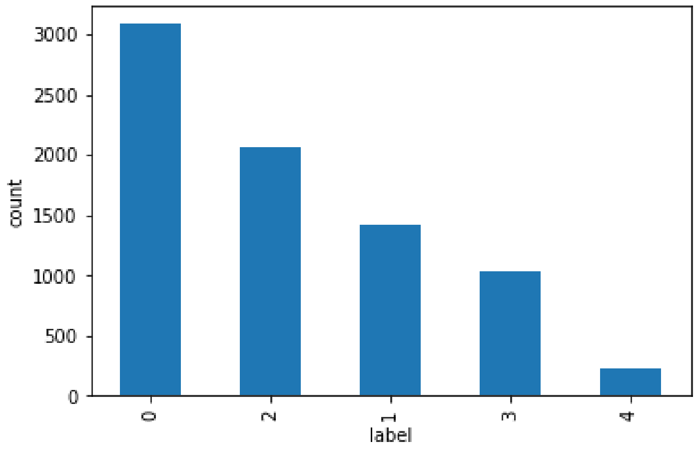
Dataset

- Training Data:
  - Total datapoints: 9786
  - Distributed across five classes
  - Split into five classes with noticeable class imbalance
- Validation and Testing:
  - Total datapoints: 7828
  - 65% allocated for validation, 35% for testing
  - Class imbalance present in validation and test data split.

Train Data Distribution

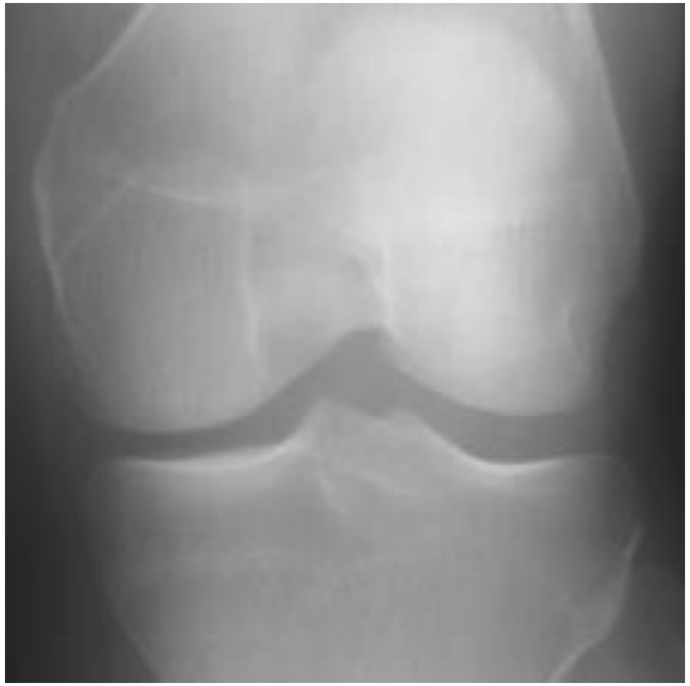


Validation and Test Data Distribution

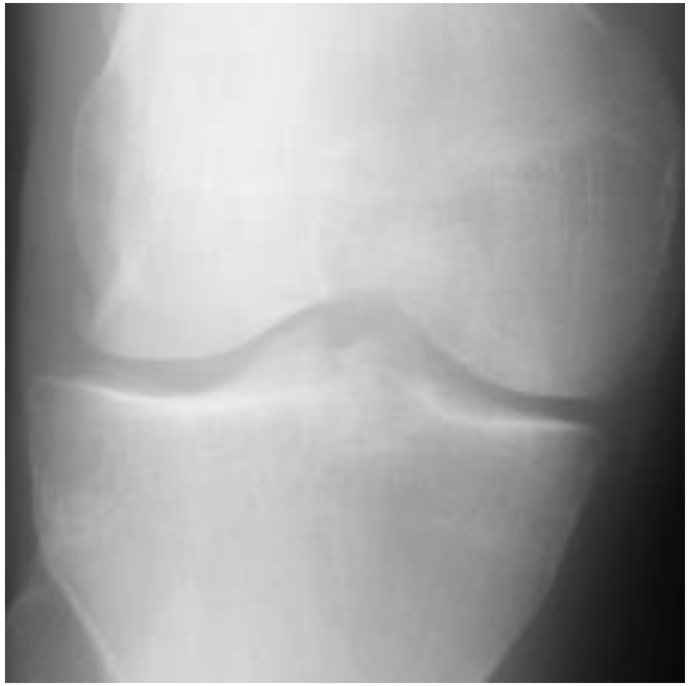


Sample X-Rays images

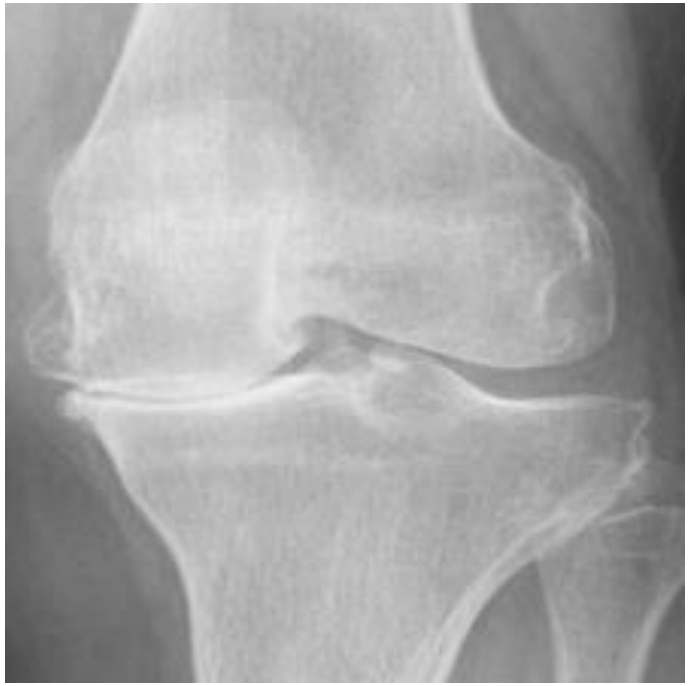
Severity class 1



Severity class 3



Severity class 5

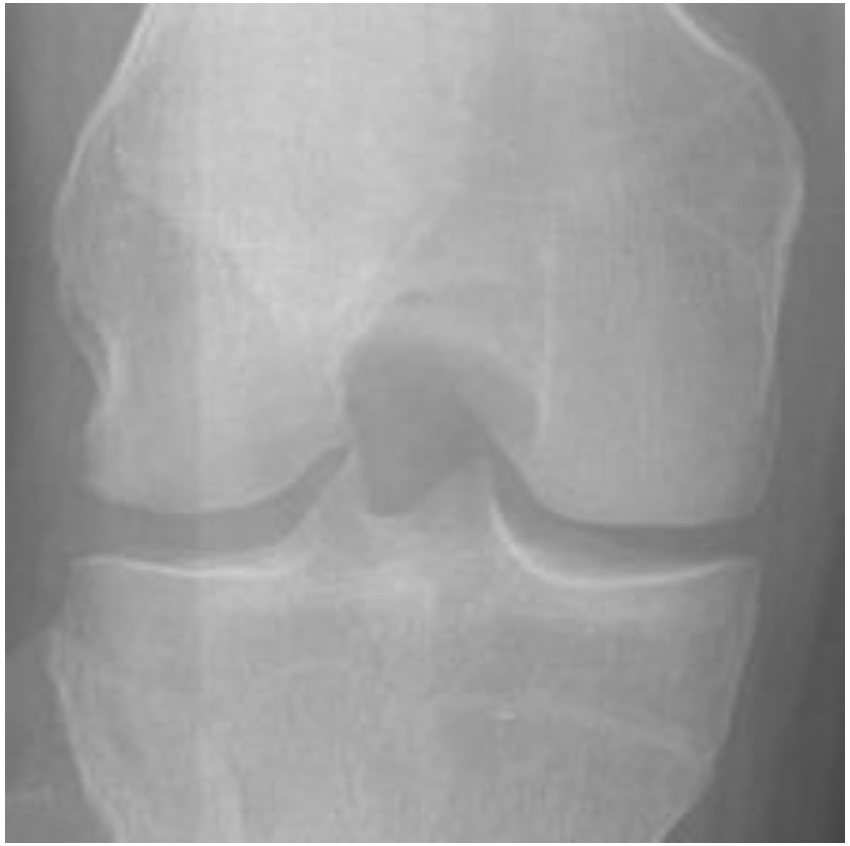


Data Pre-Processing

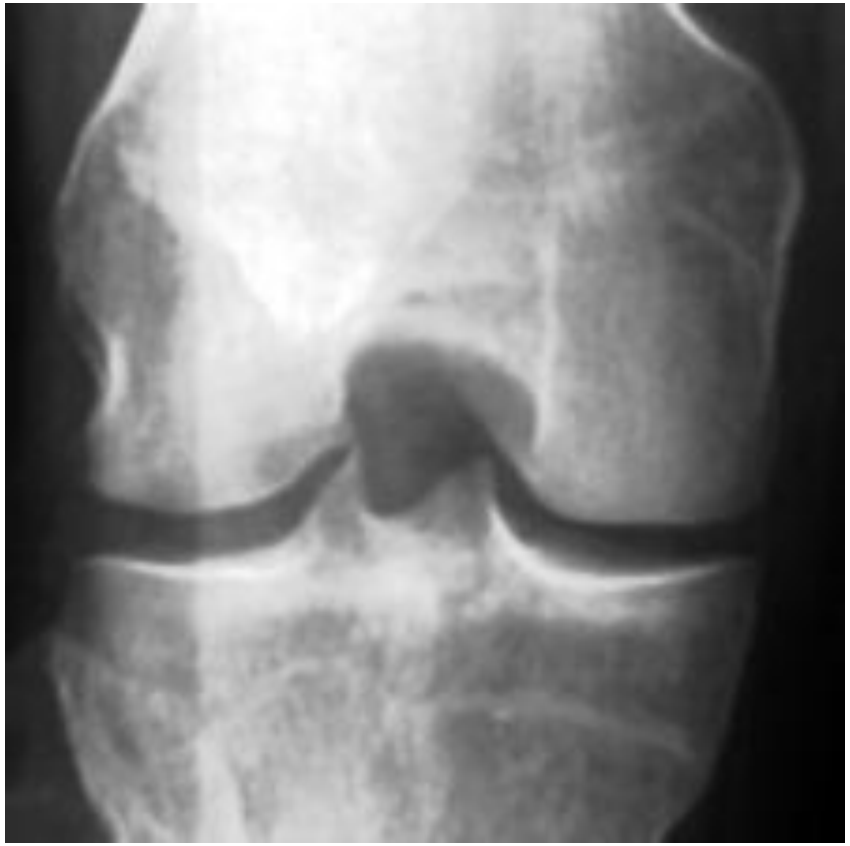
- The images underwent several steps of preprocessing to prepare them for analysis.
- Resized to a consistent dimension and converted to grayscale.
- Pixel values normalized to standardize brightness and contrast.
- Gaussian blurring applied to reduce noise.
- Histogram equalization performed to enhance the visibility of important features.

Sample Processed X-Ray Images

Original Image



Processed Image



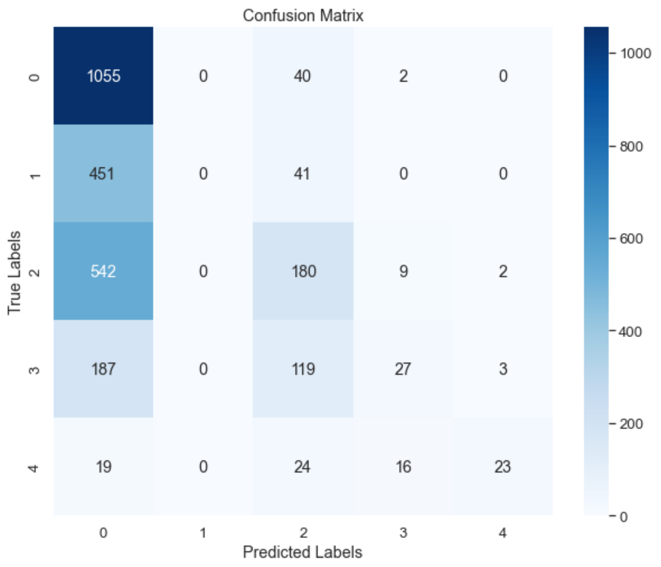
Models

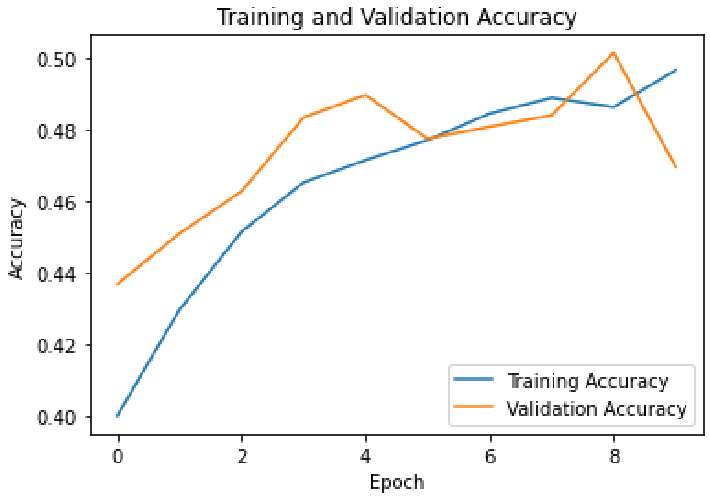
Baseline Model

Model Architecture

- Our Approach:
  - We based our project on the VGG19 convolutional neural network (CNN), renowned for image classification.
  - Transfer Learning: Utilized VGG19 with default hyperparameters.
  - Parameters: VGG19 has 20,156,997 parameters, 133,613 trainable.
- Training Strategy:
  - Pre-Trained Weights: Used VGG19's pre-trained ImageNet weights, freezing them.
  - Custom Classification: Included GlobalAveragePooling2D, dense layers with ReLU activations, and a softmax output layer.
- Optimization and Monitoring:
  - Optimizer: Employed 'adam'.
  - Loss Function: Utilized 'sparse_categorical_crossentropy'.
  - Metric: Tracked 'accuracy' during training.

Metrics





Classification Report:

	precision	recall	f1-score	support
0	0.47	0.96	0.63	1097
1	0.00	0.00	0.00	492
2	0.45	0.25	0.32	733
3	0.50	0.08	0.14	336
4	0.82	0.28	0.42	82
accuracy			0.47	2740
macro avg	0.45	0.31	0.30	2740
weighted avg	0.39	0.47	0.37	2740

VGG19 Model

Callbacks and Checkpoints

ModelCheckpoint:

Monitors validation accuracy (val_acc).

Saves model weights to 'vgg19_best.ckpt'.

Ensures only best-performing weights are saved (save_best_only=True).

Crucial for data loss protection and retaining progress.

EarlyStopping:

Halts training if validation loss (val_loss) stops improving.

Triggered after a specified number of epochs.

Prevents overfitting by terminating training when model's performance degrades.

ReduceLROnPlateau:

Dynamically adjusts learning rate during training.

Monitors validation loss (val_loss).

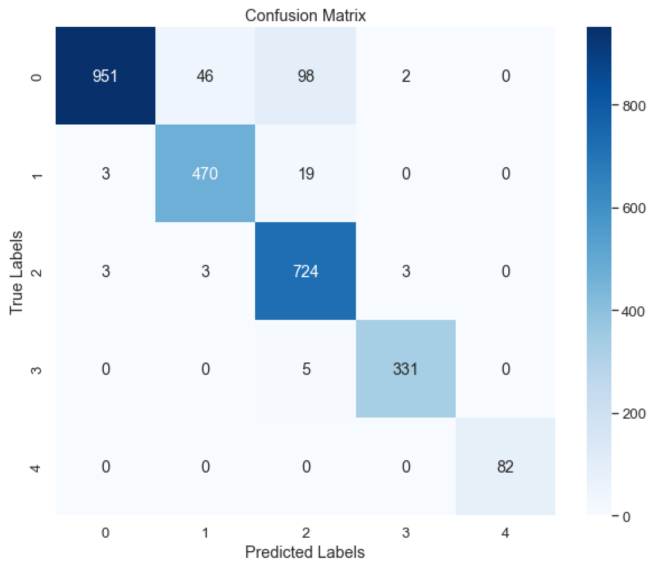
Reduces learning rate if no improvement for a certain number of epochs.

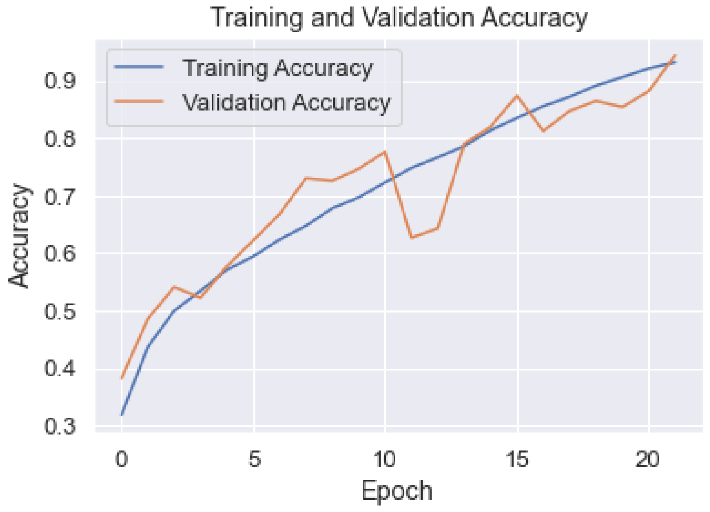
Aids in fine-tuning training process, facilitating efficient convergence and improving performance.

Model Architecture

- Transfer Learning Strategy:
    - Utilize pre-trained VGG19 model with ImageNet weights.
    - Freeze layers to focus on task-specific adaptation.
- Custom Classification Head:
    - Replaced original VGG19 head with custom architecture.
    - Four convolutional layers with batch normalization.
    - Captures dataset-specific features for improved classification.
- Class Weights for Imbalance:
    - Introduced custom weights to address class imbalances.
    - Higher weights for minority classes prevent overlooking during training.
- Optimization Techniques:
    - Adam optimizer with carefully selected learning rate (0.00001) and weight decay (0.0001).
    - Low learning rate aids fine-tuning, weight decay encourages generalization.
- Loss Function and Metrics:
    - Utilized 'sparse_categorical_crossentropy' loss for multi-class classification.
    - Tracked progress using 'accuracy' metric for percentage of correctly classified examples.

Metrics





Classification Report:

	precision	recall	f1-score	support
0	0.99	0.87	0.93	1097
1	0.91	0.96	0.93	492
2	0.86	0.99	0.92	733
3	0.99	0.99	0.99	336
4	1.00	1.00	1.00	82
accuracy			0.93	2740
macro avg	0.95	0.96	0.95	2740
weighted avg	0.94	0.93	0.93	2740

Xception Model

Callbacks and Checkpoints

ModelCheckpoint:

Monitors validation accuracy.

Saves model weights to 'xception_best.ckpt'.

Ensures only best-performing weights are saved

Crucial for data loss protection and retaining progress.

EarlyStopping:

Halts training if validation loss stops improving.

Triggered after a specified number of epochs.

Prevents overfitting by terminating training when model's performance degrades.

ReduceLROnPlateau:

Dynamically adjusts learning rate during training.

Monitors validation loss.

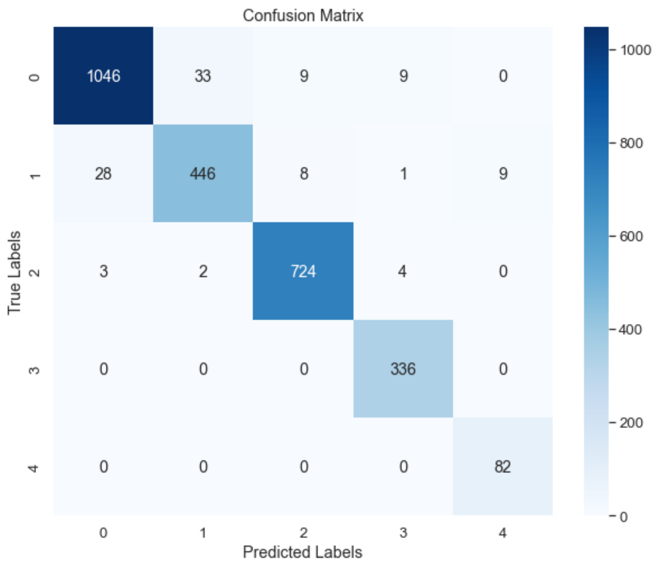
Reduces learning rate if no improvement for a certain number of epochs.

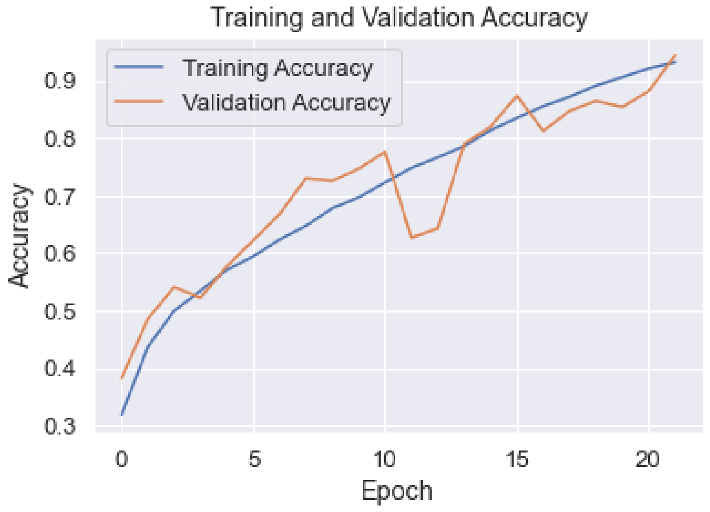
Aids in fine-tuning training process, facilitating efficient convergence and improving performance.

Model Architecture

- Depthwise Separable Convolutions:
  - Decreases computational cost and parameter count.
  - Mitigates overfitting and accelerates training.
- Refined Transfer Learning Approach:
  - Utilizes pre-trained Xception weights from ImageNet.
  - Fine-tunes for specific task while leveraging robust, generalized features.
- Custom Classification Head:
  - Replaces original head with custom architecture.
  - Includes convolutional layers with batch normalization.
  - Promotes effective extraction of dataset's unique patterns for improved performance.
- Class Weights for Imbalance:
  - Assigns higher weights to minority classes to address imbalance.
  - Prevents model from overlooking important classes during training.
- Optimization Techniques:
  - Adam optimizer with carefully selected learning rate (0.00001) and weight decay (0.0001).
  - Encourages generalization and aids fine-tuning of pre-trained model.
- Loss Function and Metrics:
  - Utilizes 'sparse_categorical_crossentropy' loss.
  - Tracks progress using 'accuracy' metric for evaluation.

Metrics

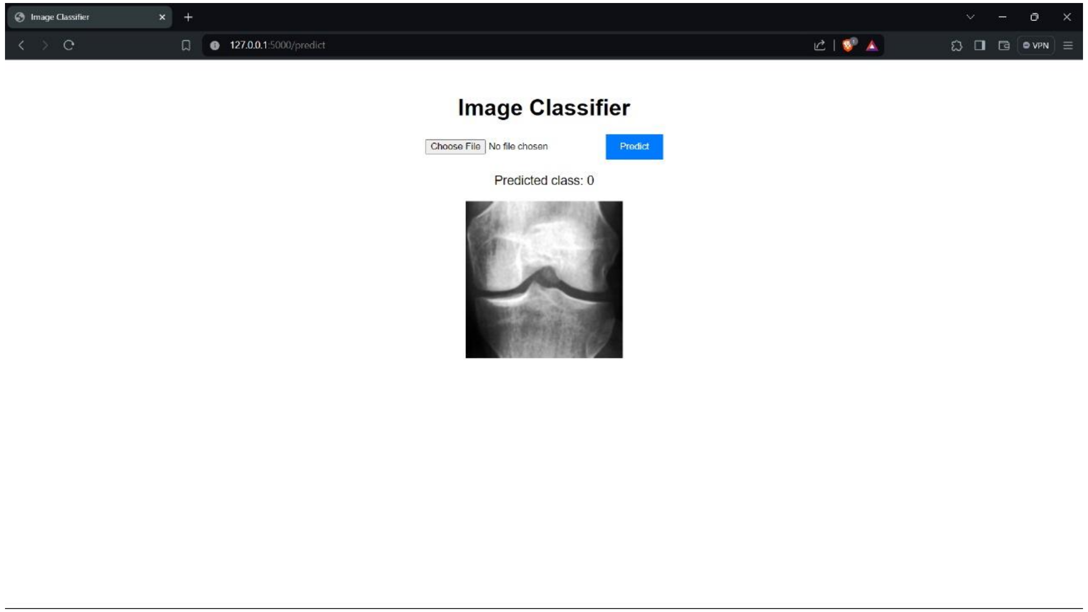


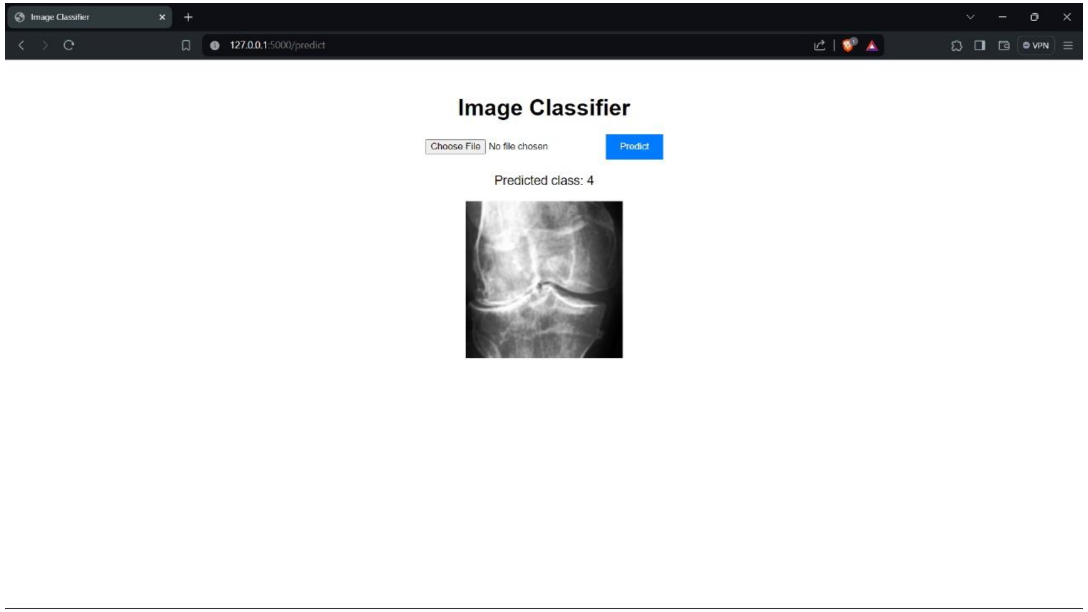



```
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.95      0.96      1097
           1       0.93      0.91      0.92       492
           2       0.98      0.99      0.98       733
           3       0.96      1.00      0.98       336
           4       0.90      1.00      0.95        82

    accuracy                           0.96      2740
   macro avg       0.95      0.97      0.96      2740
weighted avg       0.96      0.96      0.96      2740
```

Website

Website Examples





Conclusion

Conclusion

- Developed automated system for knee osteoarthritis classification using X-ray images.
- Explored effectiveness of VGG19 and Xception with customized transfer learning and class weights.
- Both models achieved promising results via careful analysis of reports and matrices.
- Xception slightly outperformed due to efficient convolutions.
- Demonstrates deep learning's value in diagnosis.
- Provides reliable, automated tool for medical image assessment.