Automatic Detection and Grading of Osteophytes on Knee Magnetic Resonance Images Using Deep Learning

T.J.F. Wolterbeek
4670582

Delft University of Technology

Abstract

Osteoarthritis (OA) is a chronic musculoskeletal joint disease that leads to disability. Osteophytes are a hallmark of OA in the knee, characterized by the formation of bone spurs that contribute to joint pain and reduced mobility. This study explores the application of deep learning (DL) techniques for the automatic detection and grading of osteophytes on magnetic resonance (MR) images of the knee. Leveraging the DenseNet-121 and ResNet-50 DL architectures from the Medical Open Network for Artificial Intelligence (MONAI) framework and a dataset, containing 1782 double echo steady-state (DESS) MR images from the Osteoarthritis Initiative (OAI), the study aims to enhance diagnostic accuracy and efficiency in medical imaging analysis. The dataset was split 8:2 for training and validation purposes, respectively. Through a series of numerical experiments, the research evaluates binary classification, region of interest (ROI)-based detection, and multi-class classification models, demonstrating that DenseNet-121 generally outperforms ResNet-50. The five-fold cross-validated binary DenseNet-121 model achieved an area under the receiver operating characteristic curve (ROC AUC) score of 0.90 and a balanced accuracy of 0.82, with a 95% confidence interval (CI) of 0.81-0.83 trained on resampled whole knee images. Moreover, the cross-validated ROI detection models for the patella inferior, superior, and tibia lateral subregions achieved balanced accuracy scores of 0.89 (0.88-0.90 CI), 0.86 (0.85-0.87 CI), and 0.85 (0.84-0.86 CI), respectively. However, the multi-class DenseNet-121 model achieved lower performance, with a balanced accuracy of 0.73 (0.71-0.75 CI), indicating the complexity of multi-class classification in this context. Furthermore, this research did not include hyperparameter optimization, as many settings were kept at their default values, suggesting the possibility for improved results. The cross-validated models were evaluated on an external test set, obtained from the Erasmus Medical Centre, comprising FSPGR-FS images from a significantly younger patient cohort, with a notable class imbalance. The models' performance on this dataset was significantly lower than their validation results, underscoring the limitations in generalizing to different age demographics and class distributions. External testing underscores the need for more robust models to maintain high performance across diverse datasets and clinical settings. Key contributions of this study include the use of weighted categorical cross-entropy (WCCE) loss functions and analysis of the knee's subregions to improve detection accuracy. The findings establish a solid foundation for further research, suggesting future work should focus on advanced optimization techniques, mixed imaging sequences in the training dataset, and comparative studies with other established models within the computer vision sector.

Automatic Detection and Grading of Osteophytes on Knee Magnetic Resonance Images Using Deep Learning

By

Ties Wolterbeek
4670582

in partial fulfillment of the requirements for the degree of

Master of Science
in Mechanical Engineering
Track Biomechanical Design

at the Delft University of Technology,
to be defended publicly on Friday, October 11, 2024, at 10:00 PM.

Thesis committee:

Dr. N.S. Tümer, chair	TU Delft
Dr. J. Hirvasniemi	Erasmus MC
Dr. J.H. Krijthe	TU Delft

An electronic version of this thesis is available at http://repository.tudelft.nl/

1 Introduction

Osteoarthritis (OA) is a chronic musculoskeletal joint disease and the most common musculoskeletal disease that leads to disability [1]. The breakdown of the joint cartilage and underlying bone can lead to pain, stiffness, and impaired movement. Since OA has no cure, physical therapies and interventions, such as weight loss, are the only methods to decrease the progression rate and to relieve the pain temporarily [2]. One of the hallmark features of OA is the formation of osteophytes, which are osteo-cartilaginous protrusions developing at the edges of the knee joint from a process that involves endochondral ossification [3, 4].

Traditionally, the diagnosis of OA and its severity assessment have relied on clinical examination and the interpretation of radiographic images acquired through different imaging modalities, such as X-ray and Magnetic Resonance Imaging (MRI). While X-ray imaging is widely used for its accessibility, low costs, and effectiveness in showing hard tissues like bone, MRI provides comprehensive details of joint anatomy, including cartilage, soft tissue changes, and bone, offering a more sensitive method for detecting early OA changes [5].

Visualizing osteophytes in MRI volumes is often done using gradient echo type sequences like dual echo steady state (DESS), spoiled gradient recalled acquisition (SPGR), or non-fat-saturated short echo time-weighted sequences like intermediate-weighted turbo spin echo (IW-TSE) [6, 3].

One of the most comprehensive and widely adopted grading criteria is the MRI Osteoarthritis Knee Score (MOAKS), a semi-quantitative (SQ) grading system proposed in 2011 by Hunter et al. [3]. The MOAKS system offers a detailed framework for evaluating the structural abnormalities associated with knee OA using MRI. This system builds upon previous grading criteria, such as the Whole-Organ MRI Score (WORMS) and the Boston Leeds Osteoarthritis Knee Score (BLOKS), by providing a more refined and nuanced approach to assessing joint pathology [3]. The MOAKS system specifically evaluates osteophytes by assessing their size and location providing a detailed and standardized measure of these bony protrusions that are characteristic of OA progression.

Despite advancements in imaging technology and grading systems, the challenges in accurately diagnosing and assessing the severity of OA have led to the recognition that more objective and efficient methods are required. The introduction of artificial intelligence (AI) into radiology offers a significant opportunity to overcome these challenges. Deep learning (DL), a subset of machine learning (ML), has shown remarkable effectiveness in tasks such as image recognition, natural language processing, and predictive analytics [7]. This effectiveness stems from the ability of DL models to learn complex patterns from extensive datasets without explicit programming.

A particular type of DL model, known as Convolutional Neural Networks (CNNs), has become dominant in the field of medical computer vision. CNNs are specifically designed to process and analyze visual data, making them highly effective for tasks involving image recognition and classification. Their architecture, which includes layers of convolutions and pooling, allows CNNs to automatically and adaptively learn spatial hierarchies of features from input images, which is crucial for accurate image analysis [7].

Specifically for knee OA, DL models are being developed to automatically detect and grade the severity of OA, identify markers that can predict the course of the disease, and predict disease progression with a level of accuracy and consistency that, in some cases, matches or even surpasses traditional manual evaluations [8, 9]. However, no studies have been identified that focus exclusively on developing a DL model specifically for osteophyte detection and grading on MRI images, using the MOAKS grading system. This gap presents a significant opportunity for future research to enhance the precision and efficiency of OA assessment through targeted DL applications. To address this gap, the aim of this study was to train and evaluate DL models, utilizing well-known CNN architectures, that could detect and grade MRI images for osteophytes, using the MOAKS grading criteria.

2 Data

2.1 Dataset

The dataset used in this study was obtained from the Osteoarthritis Initiative (OAI), which is a multi-center ten-year observational study of men and women [10]. This study aimed to provide resources to better understand the prevention and treatment of knee OA. The specific dataset utilized is from the OA Biomarkers Consortium FNIH Project within the larger OAI study, containing double echo steady state (DESS) MRI images of 600 different patients, 353 (59%) women and 247 (41%) men, at three different time points: baseline, 12 months, and 24 months, which were all combined to form the initial dataset containing 1800 images [11, 12]. The age range of the patients is between 45 and 79 years old [13]. In this project, the images were graded by two musculoskeletal radiologists with 13 and 15 years of experience, using the SQ-MOAKS grading criteria [14]. The image volumes all have the same number of voxels and voxel spacing, resulting in an image dimension of 384x384x160 voxels and a voxel spacing of approximately 0.365x0.365x0.7 millimeters. The MOAKS grades of 18 patients in the 12-month follow-up examination were incomplete and thus excluded from the study. This resulted in a

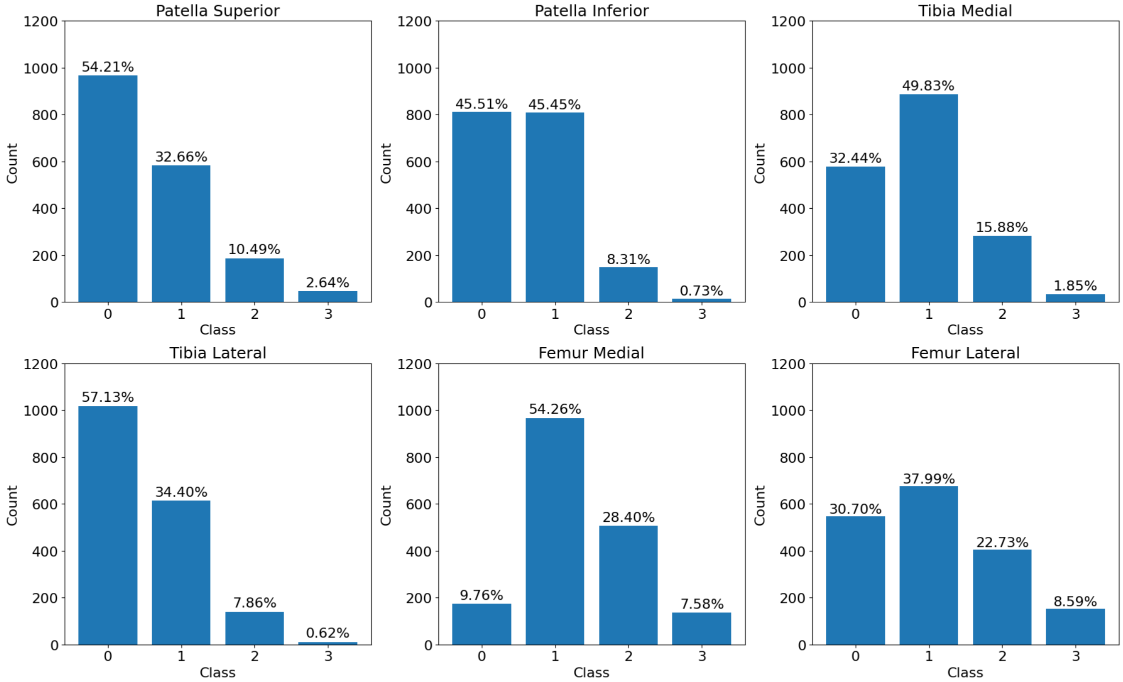


Figure 1: Class distribution ratios for each subregion of the knee. The bar graphs represent the prevalence of each class (0, 1, 2, and 3) in the patella superior, patella inferior, tibia medial, tibia lateral, femur medial, and femur lateral regions.

dataset with 1782 DESS MRI volumes and corresponding labels, that were fit for training and validation purposes.

Each image was graded for osteophytes based on the SQ-MOAKS criteria, with values ranging from 0 to 3. A grade of 0 signifies the absence of osteophytes, grade 1 denotes the presence of small osteophytes, grade 2 indicates medium-sized osteophytes, and grade 3 corresponds to large osteophytes. In the paper by Hunter et al. [3], 12 subregions of the knee are outlined in which the osteophytes are graded. However, some of these subregions have been combined for this study to create a more practical classification system. These subregions include the patella superior, patella inferior, femur lateral, femur medial, tibia lateral, and tibia medial subregions. The contracted subregions are the femur medial region, which is a combination of the femur medial anterior, femur medial posterior, and femur medial central regions, and the femur lateral region, which similarly combines the femur lateral anterior, femur lateral posterior, and femur lateral central. Combining these subregions facilitated a more practical local analysis of osteophytes. This approach preserved the crucial pathological differences across various subregions while minimizing the number of boundaries and overlap between regions, thereby simplifying the classification system and resulting in more manageable local analyses.
A few other adjustments and additions were made to the image labels. A maximum osteophyte score was added, which effectively took the largest osteophyte score of all the subregions, providing a single label value for an image. Furthermore, a binarized version of the maximum osteophyte score was added, where a maximum osteophyte score of 0 or 1 got put into class 0, and a maximum score of 2 or 3 got put into class 1. The distribution of this label was almost equal and the bar graph of the exact distribution can be found in the Appendix in Figure 14. Additionally, the image labels were further enhanced. Each subregion received a binary score, based on its local osteophyte score, with a threshold score of ≥ 1 to be categorized in the positive class. The new binarized class distribution can be seen in Figure 2.

An external test set, obtained from the Erasmus Medical Centre (EMC) in Rotterdam, the Netherlands, was utilized to objectively evaluate the proposed models on new, different data [15]. This test set included 136 MRI images, obtained using a fast spoiled gradient-echo fat-suppressed (FSPGR-FS), a sequence that provides excellent soft tissue contrast and suppresses fat tissues [16]. The ages of the patients ranged from 14 to 40 years old, with a mean age of 23. The images each had a spatial dimension of 512x512x216 voxels and a voxel spacing of 0.293x0.293x0.5 millimeters. After removing corrupt images, images without grading, or images without correct segmentation masks, 132 images remained. However, the OAI training set consisted of patients aged 45 to 79 years,

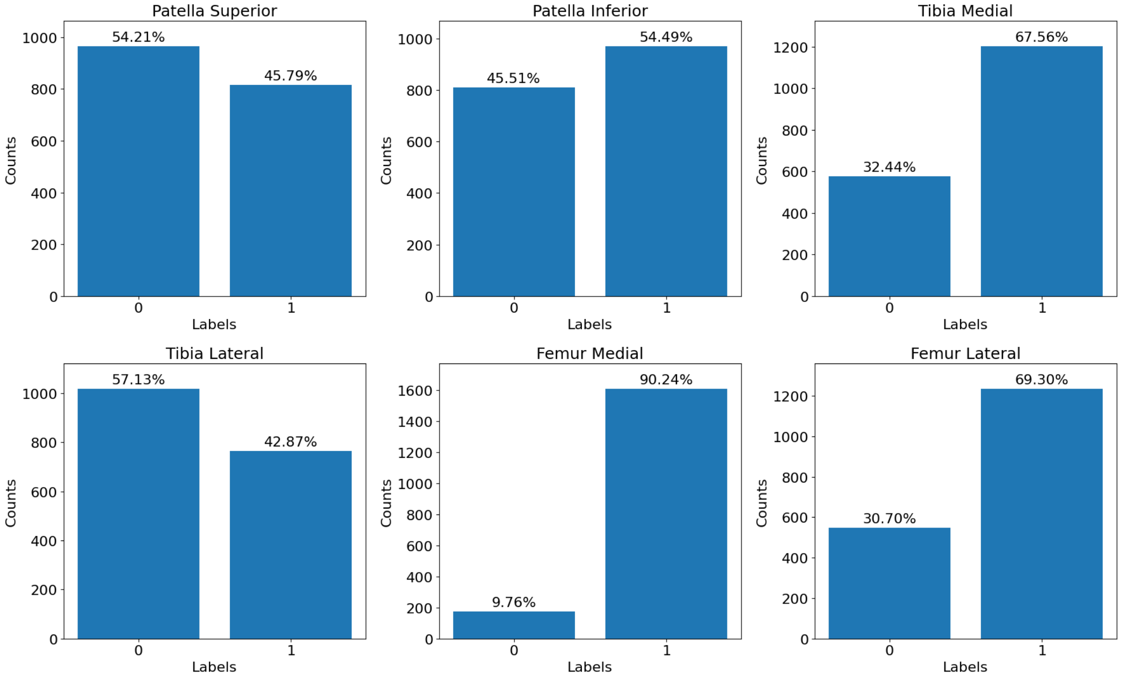


Figure 2: Binary class distribution ratios for each subregion of the knee. The bar graphs represent the prevalence of each class (0 and 1) in the patella superior, patella inferior, tibia medial, tibia lateral, femur medial, and femur lateral regions.

compared to the test set's age range of 14 to 40. Younger patients typically have more red bone marrow visible in the cancellous bone, which appears lighter on MRI scans compared to older patients who have more yellow marrow due to the replacement of hematopoietic tissue with fat over time in the metaphysis and epiphysis parts of long bones [17]. This difference is evident in the MRI images, where parts of the femur and tibia of younger patients appear lighter (Figure 3) compared to those of older individuals. Therefore, the images of patients younger than 18 were excluded, leaving 93 images to be used for the test set. The mean age became 26. These images followed the same preprocessing steps as the images from the OAI dataset and were resampled to the input dimensions each model was trained on, to present this data uniformly. The dataset balance of all classes is presented in Figure 4.

2.2 Dataset split

The dataset of 1782 DESS MRI volumes was split into a training and validation set, containing 80% and 20% of the data, respectively. The training set is utilized to train the model, enabling it to identify patterns and characteristics of the data. The validation set is reserved and only used for assessing the model's generalization capability, offering an unbiased evaluation of the model's performance on new, unseen data.

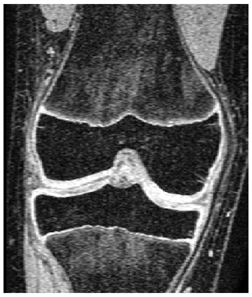


Figure 3: Coronal MRI slice of a 14-year-old patient's knee joint. The lighter appearance of the femur and tibia bones is due to the presence of red bone marrow, which is more prevalent in younger individuals and provides a higher signal intensity on MRI scans compared to the yellow marrow found in older patients.

3 Methods

3.1 Overview

This section details the various experiments conducted in this study to develop and evaluate DL models for the detection and grading of osteophytes. All networks were trained using an RTX 2080 Ti 11GB GPU with 256 GB of RAM, or a P6000 24GB VRAM GPU. All best-performing models were cross-validated, using a five-fold cross-validation method. The Adam optimizer was used to train every model, with a standard learning rate of 0.001.

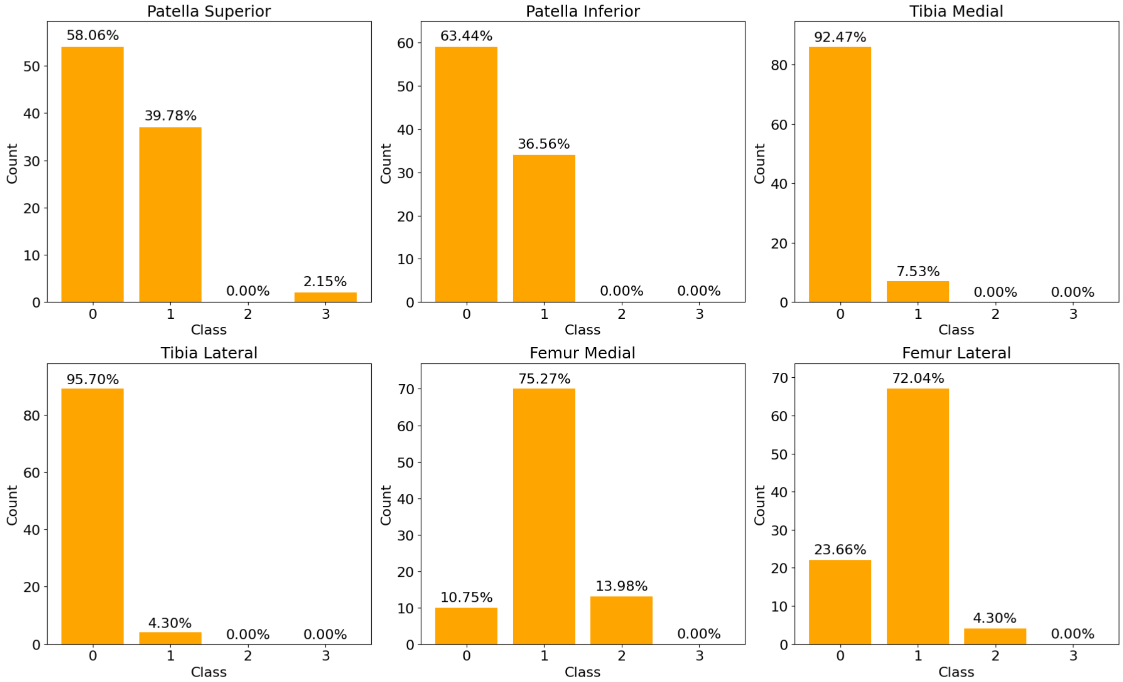


Figure 4: Multi-class distribution ratios for each subregion of the knee for the EMC test set. The bar graphs represent the prevalence of each class (0 and 1) in the patella superior, patella inferior, tibia medial, tibia lateral, femur medial, and femur lateral regions.

The dropout rate was set at a value of 0.1. Two Medical Open Network for Artificial Intelligence (MONAI) network architectures were utilized: a 3D DenseNet-121 and a 3D ResNet-50 [18]. These architectures will be further explained in subsection 3.3.

Both binary and multi-class classification models were developed, as the multi-class model aimed to directly assign MOAKS grades, while the binary model focused on detecting the presence of osteophytes. Although the multi-class approach offers a more detailed analysis, the binary model provides a simpler, yet valuable, tool to flag areas of concern, thus also aiding in the initial screening process.

3.1.1 Experiment 1: Binary resampled whole knee osteophyte classification

In this experiment, a binary classification model was developed to detect osteophytes in resampled whole knee MRI images, to test the ability of a MONAI model with default hyperparameters to detect the presence of osteophytes of MOAKS grade 2 or higher. The first step in this experiment was comparing the early results of a ResNet-50 and a DenseNet-121 model, to determine which performed better and to eventually use for final training. Due to the computational demands, the training was limited to 30 epochs, as even this preliminary phase required approximately 3 days of training time. The best-performing model was trained for 60 epochs.

3.1.2 Experiment 2: ROI-based osteophyte detection

The model setup from Experiment 1 was taken to train six models for detecting osteophytes with a MOAKS grade of 1 or higher in the six predefined ROIs. Each model was trained for 150 epochs.

3.1.3 Experiment 3: Multi-class classification on most balanced subregion

For this experiment, the subregion with the most balanced class ratio, the lateral side of the femur, was used to train multi-class classification models. Multiple loss functions, including categorical cross-entropy (CCE), weighted categorical cross-entropy (WCCE), and focal loss, were tested to find the most effective one for improving model performance. These loss functions are discussed in detail in subsection 3.4. Expanding this experiment, the best-performing loss function, with corresponding model architecture, was utilized to compare the initial learning rate value with learning rates of 0.01 and 0.0001. Given the adaptive nature of the Adam optimizer [19], learning rate adjustments were made in powers of ten rather than fine-tuning with small adjustments, to observe significant variations in training dynamics. Additionally, a model was trained with a dropout rate of 0.5 to determine if there was a significant difference in performance compared to the standard dropout rate. This approach aimed to assess the impact of increased regularization on the model's robustness and generalization capabilities. Furthermore, the best-performing loss function model setup was trained on the resampled versions of the cropped input images, which had half the number of voxels in every direction compared to the initial cropped image, to evaluate its performance and training time. Lastly, the model with the best performance metrics got five-fold cross-validated.

3.1.4 Experiment 4: Testing on an external test set

All cross-validated models were tested on the external test set from the EMC. The same configurations of the models were used as in the other experiments.

3.2 Preprocessing

To enhance the quality and consistency of the images, several preprocessing techniques were applied. First, segmentation masks that depict the patella, tibia, femur, femoral and tibial cartilage, and Hoffa's fat pad were acquired for each image, using a segmentation DL model developed by Campos et al. [20]. This segmentation model utilizes the nnU-Net framework, which automatically adapts its architecture, preprocessing, and training strategies to best suit the data it's applied to. This framework dynamically adjusts based on the dataset's characteristics, such as image resolution, contrast, and segmentation task specifics.

Each specific tissue's segmentation mask has a unique integer voxel intensity value ranging from 1 to 6. These segmentation masks were used to identify the dimensions of the cropping bounding boxes, extracting each subregion from the original image for individual evaluation. For the patella, the dimensions of the bounding boxes were determined by selecting the largest values from three different patella masks in the x, y, and z directions, respectively, where the x-axis is oriented along the anterior-posterior direction, the y-axis along the superior-inferior direction, and the z-axis along the medial-lateral direction. This means that the final bounding box was defined by taking the maximum extent from the set of masks along each individual axis. The same was done for the tibia and femur, however, in these cases the femoral and tibial cartilage masks were also included in determining the bounding box dimensions, to ensure no potential spatial information loss during cropping. To limit these bounding boxes to the medial or lateral compartments, the bounding box was divided at the midpoint along the z-axis. Furthermore, the y-axis values are given predetermined values, such that only the condyles of the tibia and femur were evaluated. To limit the bounding box of the patella to superior and inferior sides, the y-axis value of its bounding box was halved. Finally, a buffer of 10 voxels is added in every direction to ensure that all possible relevant spatial information is included. The dimensions of the final three bounding boxes that were used for extracting the ROIs are 96x87x95 voxels for the patella, 213x110x76 voxels for the tibia, and 248x135x81 voxels for the femur.

After cropping, two or three preprocessing steps were applied to enhance data quality, depending on whether the image was of a left or right knee. The z-axis of left knee images was mirrored to match the orientation of right knee images, providing the model with uniform input data. First, the images were scaled in intensity, converting them to 8-bit unsigned integers. The intensity values were mapped between the 0^{th} and 95^{th} percentiles to a range from 0 to 255, clipping the values outside this range. This helped remove bright imaging artifacts in the original images and thus enhanced contrast, simultaneously reducing the computational load of the GPU by simplifying the data, which led to faster processing and lower memory requirements. Second, the intensity values were normalized across the dataset by subtracting the mean and dividing by the standard deviation, ensuring consistent intensity distribution for each image. This ensured that the voxel values had a mean of zero and a standard deviation of one, which helped reduce the effect of varying lighting conditions and enhanced the performance of subsequent analysis. The preprocessing steps for a left knee example image are shown in Figure 5.

Furthermore, in Experiment 1 and 4, the images were resampled to reduce the computational load. This had the most impact on the computational load for whole knee evaluations, as these images are significantly larger than the images of cropped subregions. Specifically, for the whole knee osteophyte analysis, the images were resampled to a size of 192x192x80 voxels, effectively halving the spatial dimensions in each axis. This resampling process, achieved using spline interpolation, not only reduced the number of voxels by a factor of eight but also preserved essential features necessary for accurate osteophyte analysis. By optimizing the data size without significant loss of critical information, the model's performance remained robust, while the overall processing time and resource usage were minimized. The model's training time per image was reduced from approximately 60 seconds per image to approximately 3 seconds per image. An example of the difference between an original image and its resampled version is presented in Figure 6.

Data augmentation was applied to increase the generalizability and robustness of the model. Almost every image in the dataset underwent random transformations during training, including flipping over random axes, random rotations ranging between 0°and 15°, and intensity scaling. These augmentations helped to create a diverse set of train-

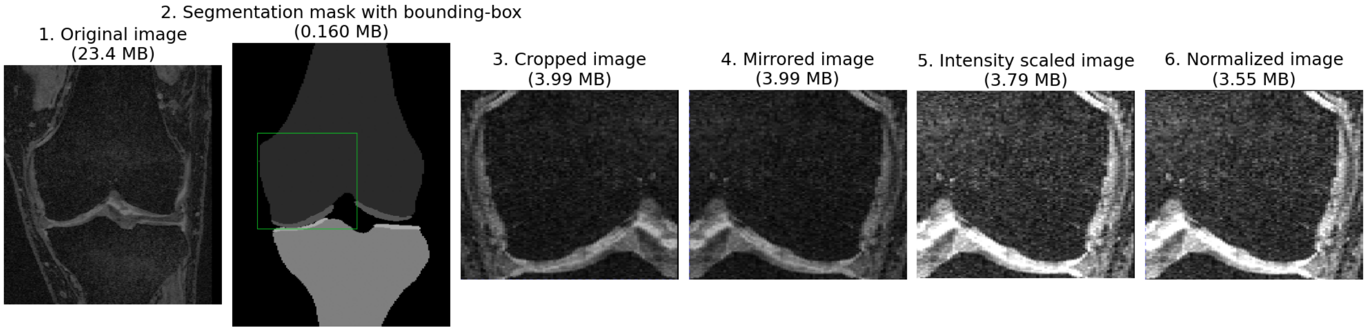


Figure 5: A 2D representation of all the preprocessing steps for a left knee image, used in a subregion analysis model. First, the bounding boxes are determined based on the segmentation mask of the input image. Then the image is cropped, mirrored, scaled in intensity, and normalized.

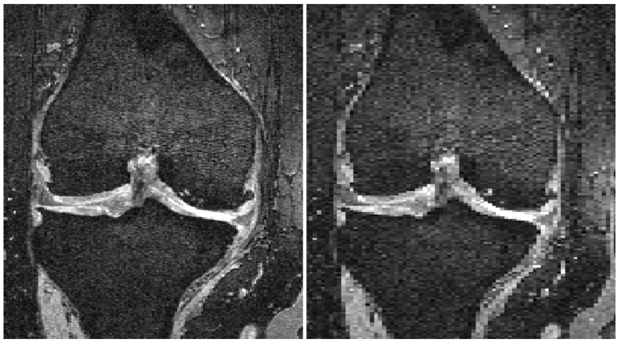


Figure 6: Example of the difference between a resampled image and its original. The amount of voxels in each direction has been halved, effectively reducing the spatial dimensions from 384x384x190 to 192x192x80 voxels.

ing examples, enabling the model to learn more invariant features and reducing overfitting. The augmentation process ensures that the model is exposed to various scenarios, thereby improving its ability to generalize well to new, unseen images, while also expanding the dataset on which it trains.

3.3 Network architectures

The MONAI DenseNet-121 model architecture consists of an initial 7x7x7 convolutional and max pooling layer, four dense blocks, each containing multiple layers, three transitional layers, and two output layers. In the dense blocks, each layer receives input from all previous layers within the block, promoting feature reuse and improving gradient flow. The dense blocks consist of 6, 12, 24, and 16 dense layers for the first, second, third, and fourth dense blocks, respectively. Every dense layer comprises a batch normalization, a rectified linear unit (ReLU), a 1x1x1 convolution operation, followed by batch normalization, another ReLU, and a 3x3x3 convolution in this order. These operations are applied to the input feature map, and the output is concatenated to the feature map. Consequently, the number of feature maps in the feature space increases by 32 with each dense layer. In between the dense blocks are transitional layers, consisting of a 1x1x1 convolutional operation and a max pooling operation, reducing the spatial dimensions of the feature maps. The output layers consist of a global average pooling layer, used to generate the final feature map, and a fully connected layer that outputs a prediction class [21]. A schematic overview of the structure of a DenseNet-121 can be seen in Figure 7.

A MONAI ResNet-50 model architecture consists of an initial 7x7x7 convolutional layer with a stride of 2, followed by a max pooling layer. The heart of the ResNet-50 architecture consists of four bottleneck blocks containing 3, 4, 6, and 3 layers respectively. Within each bottleneck block, there are a series of convolutional layers: first, a 1x1x1 convolution to reduce the number of dimensions, then a 3x3x3 convolution, followed by another 1x1x1 convolution to restore the dimensions. These layers are combined with batch normalization and ReLU activations. The distinguishing feature of a ResNet architecture is its shortcut connections, which bypass one or more layers and directly add the input to the output of the following layers. The feature maps within the bottleneck blocks grow from 64, 128, 256, to 512. Following the bottleneck blocks, a global average pooling layer is applied to create the final feature map, which is then transferred through a fully connected layer to produce the prediction class [23].

For binary classification tasks, the fully connected layer uses a sigmoid activation function to output a probability between 0 and 1, indicating the likelihood of the positive class. For multi-class classification tasks, the fully connected layer uses a softmax activation function to output a probability distribution over multiple classes. The softmax function ensures that the sum of the output probabilities for

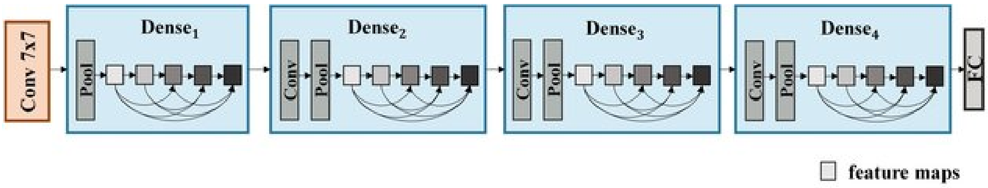


Figure 7: A schematic overview of a DenseNet-121 model architecture. It starts with an initial 7x7 convolutional layer followed by four densely connected blocks (Dense$_1$ to Dense$_4$), each consisting of multiple convolutional and pooling layers where each layer within a block is connected to all preceding layers to maximize feature reuse and learning efficiency. Transition layers between the dense blocks reduce the spatial dimensions of the feature maps. Finally, a fully connected layer processes the combined features from the dense blocks for classification into 4 classes. Image by [22].

all classes equals 1, facilitating the selection of the most likely class based on the highest probability.

3.4 Loss functions

Several different loss functions have been used during training. These loss functions will be described in this subsection.

3.4.1 Binary cross-entropy loss

Binary cross-entropy loss, also known as log loss, is utilized for classification tasks where the model needs to predict one of two output classes. The equation can be seen below.

$$\begin{aligned}\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \cdot \log(p_i) \\ +(1-y_i)\cdot\log(1-p_i)]\end{aligned} \tag{1}$$

Here $\mathcal{L}_{BCE}$ is the loss for a single instance, N is the number of data samples, y_i is the actual label of sample i (which can be either 0 or 1), p_i is the predicted probability that sample i belongs to class 1. The formula computes the loss for each instance by taking the actual label y_i, and if y_i is 1, it uses the log of the predicted probability p_i. If y_i is 0, it uses the log of $1-p_i$ (the predicted probability of class 0). It then averages the loss across all N samples. The negative sign in front ensures that the loss is a positive number.

3.4.2 Categorical cross-entropy loss

In a multi-class classification setting, where you have more than two possible classes, the cross-entropy loss function is extended to what is known as the categorical cross-entropy loss. For each instance in your dataset, the model will output a probability for each class, indicating how likely it thinks the instance belongs to that class. The categorical cross-entropy loss function then compares these predicted probabilities with the true labels.

$$\mathcal{L}_{CCE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\cdot\log(p_{ij}) \tag{2}$$

Here M is the number of classes, and p_j is the predicted probability of the observation belonging to class j. For a given instance, the loss is calculated by taking the log of the predicted probability for the true class and multiplying it by -1. If the predicted probability is high (close to 1), the log value is closer to 0, and the loss is low. If the predicted probability is low (far from 1), the log value is a large negative number, and when multiplied by -1, results in a high loss. The overall loss for a batch or the entire dataset is typically the average of the loss across all instances.

In cases where there is an imbalance in the dataset, with certain classes being significantly more common than others, a weighted categorical cross-entropy loss function can assign greater importance to the less frequent classes. This allows the model to focus more on the underrepresented classes, thereby enhancing its performance in these classes. The weighted categorical cross-entropy loss adjusts the regular categorical cross-entropy loss by incorporating a weight factor for each class. The formula for the weighted categorical cross-entropy loss is:

$$\mathcal{L}_{WCCE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} w_j\cdot y_{ij}\cdot\log(p_{ij}) \tag{3}$$

3.4.3 Focal loss

Focal loss is an adapted form of cross-entropy loss that addresses class imbalance by incorporating a modulating factor. This factor decreases the loss attributed to well-classified examples ($p_{ij} \geq 0.5$), emphasizing the learning process on hard, incorrectly classified instances. The focusing hyperparameter γ, which plays a significant role in controlling the extent to which easier examples are down-weighted, starts reducing the relative loss for well-classified examples when $\gamma > 0$ [24]. The equation for the focal loss is shown below.

$$\mathcal{L}_{FL} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M}\alpha_j \cdot (1-p_{ij})^{\gamma} \cdot \log(p_{ij}) \tag{4}$$

Here α_j is the weighting factor for class j, which can help address the class imbalance. γ is a modulating factor and the introduction of the $(1-p_{ij})^{\gamma}$ term is what differentiates focal loss from traditional cross-entropy loss, ensuring that the penalty for misclassified examples is adjusted based on how difficult they are to classify.

3.5 Performance metrics

The evaluation of DL models is an important step in their development and validation. The adoption of the appropriate performance metrics is fundamental to understanding a model's diagnostic capability and reliability. Furthermore, each metric provides a unique insight into the model's performance but provides no reliable insights on its own. This is why a combination of metrics is often utilized in research. The performance metrics employed in this study include accuracy, balanced accuracy, precision, recall, specificity, F1-score, and Cohen's Kappa score. The formulas for calculating these metrics are provided below, where TP is the true positives, TN the true negatives, FP the false positives, FN the false negatives, P_o is the proportion of instances where both raters agree, and P_e is the expected agreement, which is based on the distribution of each class. Additionally, the area under the curves (AUC) was computed for both the receiver operating characteristic curve (ROC), which graphs sensitivity versus the false positive rate, and the precision-recall curve (PR), which graphs precision versus recall. Further details on each performance metric and their score interpretations can be found in the Appendix. To evaluate the robustness of the performance metrics, 95% confidence intervals (CIs) were computed using the bootstrap resampling technique applied to the validation and external test sets. This process involved creating a new dataset by selecting 10^3 random samples with replacements and computing the metrics on this resampled dataset. Based on this, the 95% CIs were determined. For experiments involving multiple classes, the macro-average, which calculates the metric independently for each class and then takes the average without taking their size into account, was computed for metrics typically designed for binary classification problems.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{5}$$

$$Balanced\ accuracy = \frac{1}{2}(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}) \tag{6}$$

$$Precision = \frac{TP}{TP+FP} \tag{7}$$

$$Recall = \frac{TP}{TP+FN} \tag{8}$$

$$Specificity = \frac{TN}{TN+FP} \tag{9}$$

$$F1-score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{10}$$

$$\kappa = \frac{P_o - P_e}{1-P_e} \tag{11}$$

4 Results

In this section, the results of the 4 experiments will be presented in tables, showing the achieved performance metrics, and ROC AUC and PR AUC graphs of the cross-validated models. For all experiments, the loss functions, confusion matrices, and ROC AUC and PR AUC curves of non-cross-validated models are presented in the Appendix.

4.1 Experiment 1: Binary resampled whole knee osteophyte classification

The results for the models trained in Experiment 1 are presented in Table 1. When comparing the results of the two preliminary models, it is clear that the DenseNet network outperformed the ResNet network. This is evident across several key performance metrics: DenseNet achieved a training accuracy of 0.90 and a training loss of 0.057 compared to ResNet's training accuracy of 0.70 and a training loss of 0.14 and it achieved higher scores on all evaluation metrics, obtaining an accuracy and balanced accuracy score of 0.83, a precision of 0.88, a recall of 0.77, a specificity of 0.89, an F1 score of 0.82 and a kappa score of 0.66.

The next step in this experiment was to further optimize the performance of the best model configuration, the DenseNet-121 architecture, by training the model for a longer duration of 60 epochs. Extended training aimed to refine the model and align its performance with state-of-the-art osteophyte detection models. The five-fold cross-validated results are similar to the first DenseNet test model, as some of their results fall into each other 95% CIs.

The five-fold cross-validated DenseNet-121 model achieved a mean ROC AUC of 0.90 (± 0.04) and a mean PR AUC of 0.90 (± 0.04), as illustrated in the ROC and PR curves in Figure 8. The ROC and PR curves of the preliminary models are presented in Figure 26. The training loss functions and confusion matrices are presented in Figure 16 and Figure 21, respectively.

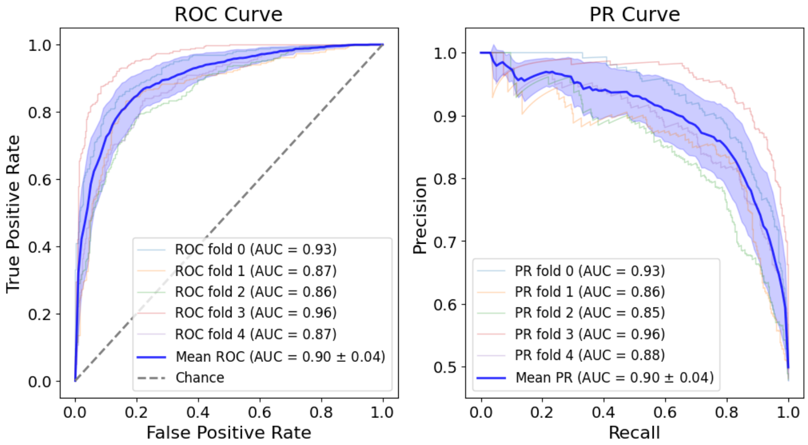


Figure 8: ROC and PR Curves for the DenseNet-121 (e = 60) Model. The ROC curve (left) and PR curve (right) display the performance of the DenseNet-121 model across five-fold cross-validation. The mean ROC AUC is 0.90 with a standard deviation of ± 0.04, indicating the model's high capability to distinguish between positive and negative classes. Similarly, the mean PR AUC is 0.90 with a standard deviation of ± 0.04, reflecting the model's precision and recall balance. The shaded areas represent the confidence intervals for each fold, demonstrating the robustness and consistency of the model's performance across different data splits.

Table 1: Results models for Experiment 1. The train and validation metrics are shown, where the validation metrics are shown on the right of the double vertical lines. The validation metrics 95% CIs are shown in the brackets, thus presenting the metrics as: mean, 95% CI [lower bounds, upper bounds]. Since the first two models were for comparing performances, only the DenseNet-121 (e = 60) model was five-fold cross-validated. Here, train stands for training metrics, A is the accuracy, BA the balanced accuracy, P the precision, R the recall, S the specificity, and κ is the Cohen's kappa metric.

Model	**Train Loss**	**Train A**	**A**	**BA**	**P**	**R**	**S**	**F1**	κ
DN-121 (e = 30)	0.057	0.90	**0.83 (0.81-0.86)**	**0.83 (0.80-0.86)**	**0.88 (0.84-0.91)**	0.77 (0.73-0.82)	**0.89 (0.86-0.92)**	0.82 (0.79-0.85)	**0.66 (0.61-0.72)**
RN-50 (e = 30)	0.14	0.70	0.77 (0.74-0.80)	0.77 (0.74-0.80)	0.78 (0.74-0.82)	0.75 (0.70-0.79)	0.79 (0.75-0.83)	0.77 (0.73-0.80)	0.54 (0.47-0.59)
DN-121 (e = 60)	0.057	0.90	0.82 (0.81-0.83)	0.82 (0.81-0.83)	0.83 (0.80-0.84)	**0.82 (0.80-0.83)**	0.83 (0.81-0.84)	**0.82 (0.80-0.83)**	0.64 (0.61-0.66)

4.2 Experiment 2: ROI-based osteophyte detection

The model architecture from Experiment 1 was retained, with the input data adjusted to cropped versions of the original image. Table 2 presents the results of each ROI model, all five-fold cross-validated. The patella inferior model achieved strong overall performance, achieving the highest balanced accuracy (0.89), specificity (0.90), and kappa score (0.79) among all the models. In contrast, the femur medial model showed strong precision (0.95) and respectable recall (0.92), though it scored the lowest on balanced accuracy, specificity and kappa score (0.64, 0.29, 0.38, respectively). The ROC and PR AUC plots are presented in Figure 9.

4.3 Experiment 3: Multi-class classification on most balanced subregion

The least imbalanced subregion was selected for multi-class classification. Table 3 shows the results of the various models, where DenseNet-121 models consistently outperformed ResNet-50 models across all loss functions. The WCCE loss function proved to be the best-performing, with the DenseNet-121 WCCE model achieving a balanced accuracy and recall of 0.87. The resampling model and the increased dropout rate model both underperformed compared to the baseline DenseNet-121 WCCE model. The resampled model had a training time of 126 hours, which was notably longer than the baseline model's 117-hour training time. Additionally, the increased learning rate model performed slightly worse than the baseline with a balanced accuracy drop of 0.07, while the decreased learning rate model achieved nearly identical metrics and was chosen for cross-validation. However, the cross-validated model showed slightly reduced performance with an accuracy of 0.76, a precision of 0.79 and a kappa score of 0.66. The ROC and PR plots are presented in Figure 10.

4.4 Experiment 4: Testing models on external test set

The results of testing the cross-validated models on an external dataset are presented in Table 4. The models performed significantly worse on the external test set than on their validation set. For instance, the whole knee resampled model achieved an accuracy of 0.80 and a specificity

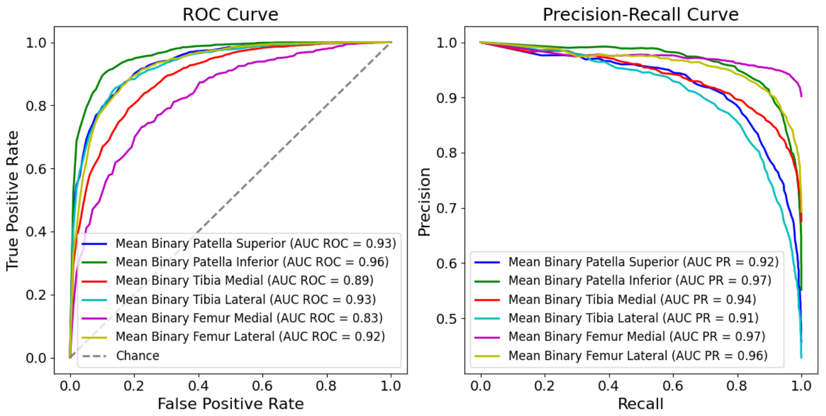


Figure 9: Mean ROC and PR Curves for all DenseNet-121 subregion models that were five-fold cross-validated. The ROC curve (left) and PR curve (right) display the performance of all 6 DenseNet-121 models. When evaluating these results, keep the dataset class distributions, presented in Figure 2, in mind.

Table 2: Results models for Experiment 2. The train and validation metrics are shown, where the validation performance metrics are shown on the right side of the double vertical lines. The validation metrics 95% CIs are shown in the brackets. All ROIs models were DenseNet-121 networks, trained for e = 150 and are five-fold cross-validated. PS and PI are the superior and inferior sides of the patella, TM and TL are the medial and lateral sides of the tibia, and FM and FL the medial and lateral sides of the femur, respectively. Furthermore, train stands for training metrics, A is the accuracy, BA the balanced accuracy, P the precision, R the recall, S the specificity, and κ is Cohen's kappa metric.

ROI	Train Loss	Train A	A	BA	P	R	S	F1	κ
PS	0.032	0.94	0.86 (0.85-0.87)	0.86 (0.85-0.87)	0.87 (0.85-0.88)	0.82 (0.81-0.84)	0.89 (0.88-0.90)	0.84 (0.83-0.86)	0.72 (0.70-0.74)
PI	0.039	0.93	0.89 (0.88-0.90)	**0.89 (0.88-0.90)**	0.91 (0.90-0.93)	0.89 (0.88-0.90)	**0.90 (0.88-0.91)**	0.90 (0.89-0.91)	**0.79 (0.77-0.81)**
TM	0.059	0.89	0.83 (0.81-0.84)	0.77 (0.75-0.78)	0.83 (0.81-0.84)	0.94 (0.93-0.95)	0.59 (0.56-0.62)	0.88 (0.87-0.89)	0.57 (0.56-0.62)
TL	0.048	0.90	0.85 (0.84-0.86)	0.85 (0.84-0.86)	0.82 (0.79-0.83)	0.86 (0.84-0.87)	0.85 (0.83-0.86)	0.84 (0.82-0.85)	0.70 (0.68-0.72)
FM	0.039	0.95	**0.93 (0.92-0.94)**	0.64 (0.62-0.66)	**0.93 (0.92-0.94)**	**1.0 (0.99-1.0)**	0.29 (0.24-0.33)	**0.96 (0.96-0.97)**	0.38 (0.35-0.46)
FL	0.046	0.92	0.86 (0.85-0.87)	0.83 (0.82-0.85)	0.90 (0.88-0.91)	0.90 (0.89-0.91)	0.77 (0.74-0.79)	0.90 (0.89-0.91)	0.67 (0.64-0.69)

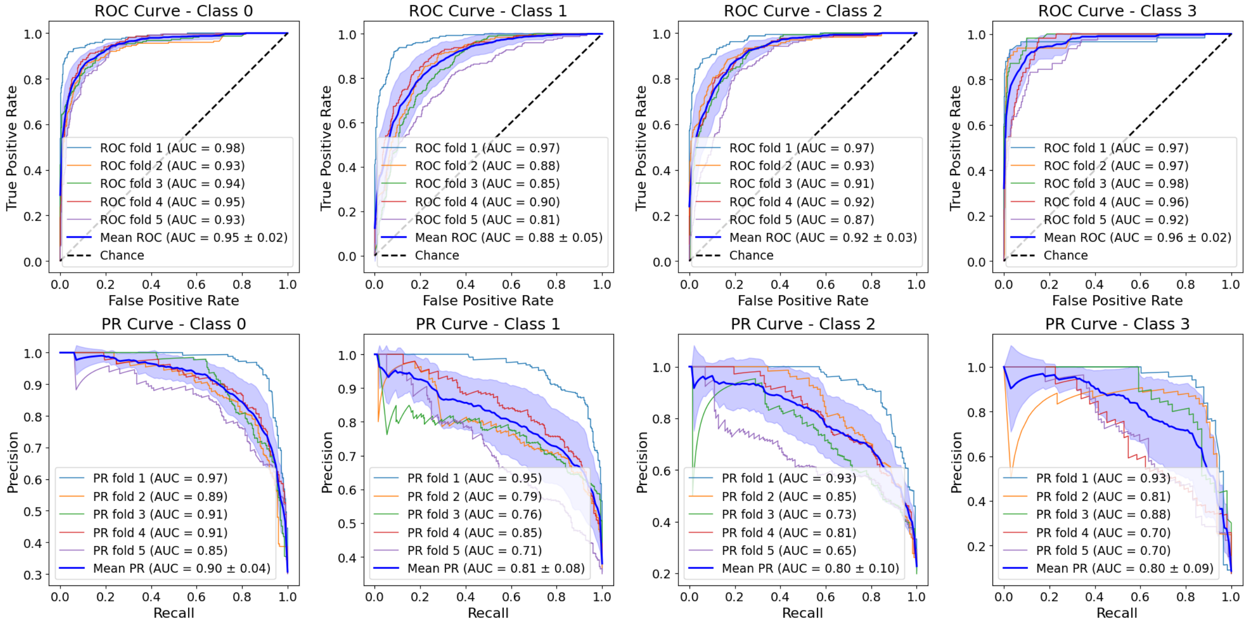


Figure 10: ROC and PR curves for the five-fold cross-validated multi-class femur lateral side model. The ROC curve (top) and PR curve (bottom) display the performance per class. Each subplot represents one class, showing how well the models distinguish between the given class and the rest. The class distribution for this subregion can be found in Figure 1.

Table 3: Results of the models for Experiment 3. The train and validation metrics are shown, where the validation performance metrics are shown on the right side of the double vertical lines. The validation metrics 95% CIs are shown in the brackets. All networks were trained for e = 150. The top six models use different loss functions. The last four models have different hyperparameters. The best-performing model (DN WCCE LR = 0.0001) was five-fold cross-validated (CV). Here A is the accuracy, BA the balanced accuracy, P the precision, R the recall, S the specificity, and κ is the Cohen's kappa metric.

Model	**Train Loss**	**Train A**	**A**	**BA**	**P**	**R**	**S**	**F1**	κ
DN-121 CCE	0.090	0.84	0.84 (0.81-0.86)	0.82 (0.79- 0.86)	0.85 (0.82-0.88)	0.82 (0.79-0.86)	0.94 (0.93-0.95)	0.84 (0.81-0.86)	0.76 (0.72-0.80)
DN-121 WCCE	0.10	0.81	**0.86 (0.83-0.88)**	**0.87 (0.84-0.89)**	0.85 (0.81-0.88)	**0.87 (0.84-0.89)**	**0.95 (0.94-0.96)**	0.85 (0.82-0.88)	0.79 (0.75-0.83)
DN-121 FL	2.28	0.64	0.81 (0.78-0.84)	0.83 (0.80-0.86)	0.81 (0.78-0.84)	0.83 (0.80-0.86)	0.93 (0.92-0.94)	0.82 (0.79-0.85)	0.73 (0.68-0.77)
RN-50 CCE	0.10	0.82	0.83 (0.80-0.86)	0.81 (0.78-0.85)	0.85 (0.81-0.88)	0.81 (0.78-0.85)	0.94 (0.93-0.95)	0.82 (0.79-0.85)	0.75 (0.71-0.79)
RN-50 WCCE	0.11	0.80	0.77 (0.74-0.81)	0.72 (0.68-0.76)	0.78 (0.74-0.82)	0.72 (0.68-0.76)	0.92 (0.91-0.93)	0.74 (0.70-0.78)	0.67 (0.63-0.72)
RN-50 FL	2.32	0.66	0.76 (0.73-0.80)	0.77 (0.73-0.80)	0.76 (0.72-0.79)	0.77 (0.73-0.80)	0.91 (0.90-0.92)	0.76 (0.72-0.79)	0.66 (0.61-0.71)
DN-121 WCCE RS	0.14	0.74	0.66 (0.62-0.69)	0.69 (0.65-0.73)	0.65 (0.62-0.69)	0.69 (0.65-0.73)	0.89 (0.87-0.90)	0.65 (0.61-0.69)	0.53 (0.48-0.58)
DN-121 WCCE DO = 0.5	0.17	0.68	0.72 (0.69-0.76)	0.73 (0.70-0.77)	0.74 (0.70-0.78)	0.73 (0.70-0.76)	0.90 (0.89-0.91)	0.73 (0.69-0.76)	0.60 (0.56-0.65)
DN-121 WCCE LR = 0.01	0.11	0.81	0.83 (0.81-0.86)	0.80 (0.76-0.84)	0.85 (0.81-0.88)	0.80 (0.76-0.84)	0.94 (0.93-0.95)	0.82 (0.78-0.85)	0.76 (0.72-0.80)
DN-121 WCCE LR = 1e-4	0.084	0.85	**0.86 (0.83-0.88)**	0.85 (0.83-0.88)	**0.87 (0.84-0.88)**	0.85 (0.83-0.88)	**0.95 (0.94-0.96)**	**0.86 (0.83-0.88)**	**0.79 (0.76-0.83)**
DN-121 WCCE LR = 1e-4 CV	0.080	0.86	0.76 (0.75-0.78)	0.73 (0.71-0.75)	0.79 (0.76-0.80)	0.73 (0.71-0.75)	0.91 (0.91-0.92)	0.75 (0.73-0.77)	0.66 (0.64-0.68)

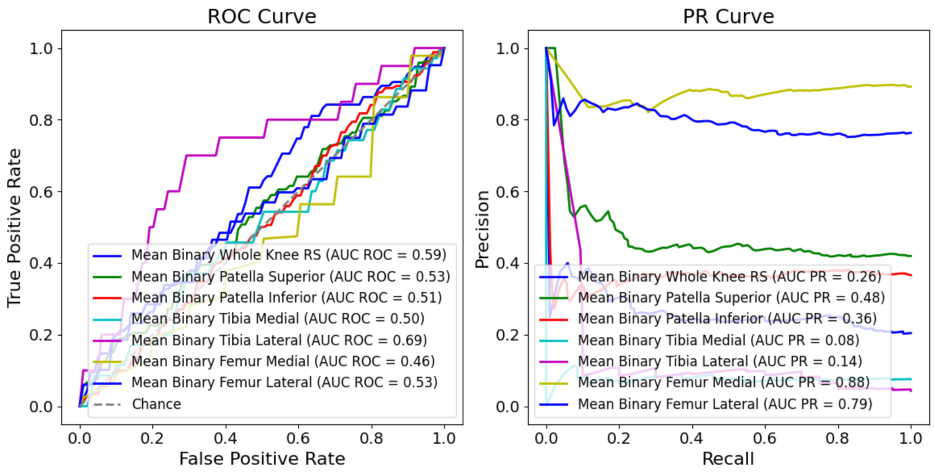


Figure 11: ROC and PR curves for all the five-fold cross-validated binary models. The class distribution for each subregion can be found in Figure 4.

Table 4: Results models for Experiment 4. The performance metrics are presented for the cross-validated models on the external test set. The validation metrics 95% CIs are shown in the brackets. All final models were DenseNet-121 networks, trained for 150 epochs, except the first whole knee model, as this was trained for only 60 epochs. The threshold for a positive class for the binary models of the subregions is a grade > 0, while for the whole knee model, the threshold is a grade > 1. PS and PI are the superior and inferior sides of the patella, TM and TL are the medial and lateral sides of the tibia, and FM and FL the medial and lateral sides of the femur, respectively. Furthermore, A is the accuracy, BA the balanced accuracy, P the precision, R the recall, S the specificity, and κ is the Cohen's kappa metric.

Model	**A**	**BA**	**P**	**R**	**S**	**F1**	κ
Whole knee RS (e=60)	0.80 (0.76-0.83)	0.55 (0.51-0.58)	0.45 (0.29-0.70)	0.13 (0.065-0.20)	**0.97 (0.95-0.98)**	0.18 (0.11-0.30)	**0.12 (0.041-0.22)**
Binary PS	0.59 (0.55-0.64)	0.52 (0.50-0.55)	0.65 (0.41-0.79)	0.092 (0.053-0.14)	0.96 (0.93-0.98)	0.16 (0.095-0.23)	0.054 (0.00056-0.11)
Binary PI	0.59 (0.54-0.63)	0.49 (0.46-0.52)	0.32 (0.22-0.44)	0.12 (0.076-0.17)	0.86 (0.82-0.90)	0.18 (0.12-0.24)	-0.022 (-0.092-0.050)
Binary TM	0.55 (0.51-0.60)	0.46 (0.37-0.54)	0.061 (0.027-0.096)	0.34 (0.18-0.50)	0.57 (0.53-0.62)	0.10 (0.047-0.16)	-0.027 (-0.082-0.023)
Binary TL	0.84 (0.81-0.87)	**0.56 (0.46-0.66)**	0.090 (0.017-0.15)	0.25 (0.062-0.45)	0.87 (0.84-0.90)	0.13 (0.026-0.22)	0.070 (-0.033-0.15)
Binary FM	**0.85 (0.82-0.89)**	0.52 (0.48-0.57)	**0.90 (0.87-0.93)**	**0.94 (0.92-0.97)**	0.10 (0.021-0.19)	**0.92 (0.90-0.94)**	0.053 (-0.047-0.19)
Binary FL	0.46 (0.42-0.50)	0.52 (0.47-0.57)	0.80 (0.72-84)	0.40 (0.35-0.45)	0.63 (0.54-0.72)	0.50 (0.48-0.58)	0.015 (-0.045-0.088)
Multi-class FL LR = 0.0001	0.39 (0.34-43)	0.35 (0.32-0.38)	0.33 (0.23-0.36)	0.33 (0.24-0.38)	0.76 (0.74-0.78)	0.25 (0.18-0.29)	0.030 (-0.024-0.087)

of 0.97, but was accompanied by low recall, F1, and kappa scores of 0.13, 0.18, and 0.12, respectively. Looking at the ROI models, the binary femur medial model stands out with the highest accuracy (0.85), precision (0.90), recall (0.94), and F1 score (0.92). However, it scored the lowest specificity score (0.10). Additionally, the multi-class model achieved an accuracy and balanced accuracy of 0.39 and 0.35, respectively, with a precision and recall score of 0.33, a specificity of 0.76, and a kappa score of 0.03. The ROC and PR curves are presented in Figure 11 and Figure 12.

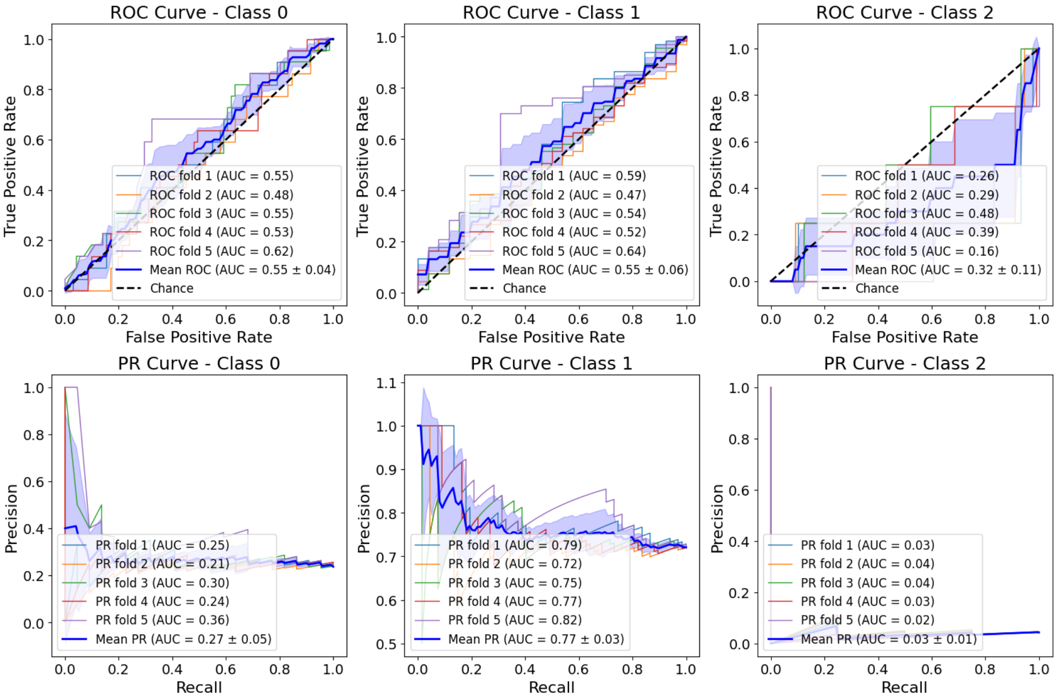


Figure 12: ROC and PR curves for the multi-class five-fold cross-validated femur lateral model. The ROC curve (top) and PR curve (bottom) display the performance per class. Each subplot represents one class, showing how well the models distinguish between the given class and the rest. Since there were no images present in the external test set of class 3, this class has been excluded from the plots. The class distribution for this subregion can be found in Figure 4.

5 Discussion

This study has made significant strides in the automatic detection and grading of osteophytes on MRI images using DL through four key experiments. Before addressing the limitations of this study per experiment, it is important to consider similar research in the field. Some of the most relevant studies have been selected as benchmarks for comparison, including assessments against manual radiologist grading using the MOAKS system.

To benchmark the results of the models in this study with manual grading results of radiologists, the results of the study by Hunter et al. [3] provide a useful point of comparison. In Hunter et al.'s study, the intra-rater kappa values for osteophyte grading ranged from 0.64 in the femoral region to 0.84 in the patella, while inter-rater kappa values were lower, ranging from 0.49 in the tibial region to 0.80 in the femoral region. The kappa scores from the DL models in this study, such as 0.79 for the patella inferior and 0.57 for the tibia medial region, closely align with the intra-rater reliability reported by Hunter et al. However, these results focus on binary classification, while Hunter et al.'s study examines multi-class classification. Notably, the femur lateral side cross-validated multi-class model achieved a kappa of 0.66, which is 0.21 lower than Hunter et al.'s findings. In contrast, the non-cross-validated model achieved a more comparable kappa score of 0.79.

Tiulpin et al. (2018) [25] conducted a study using an ensemble of two squeeze-excitation ResNet-50 networks to detect and grade the severity of whole knee OA from posterior-anterior X-rays. The models were trained on a dataset of 19,704 knees from the OAI using the Kellgren-Lawrence (KL) system and the Osteoarthritis Research Society International (OARSI) atlas for grading OA features such as osteophytes, joint space narrowing, and sclerosis, where a grade of 2 or higher indicates the presence of OA, similar to how this study classified knees with MOAKS grade 2 or 3 osteophytes as positive for osteophytes in Experiment 1. Their study reported a weighted kappa of 0.82 for predicting KL grades, and a range of 0.79 to 0.94 for specific OA features like femoral and tibial osteophytes, with an AUC of 0.98 for detecting radiographic OA. Their model was validated against the expert readings of musculoskeletal radiologists, showing strong alignment with human experts in the detection and grading of OA features. While Tiulpin et al. assessed multiple OA features, this study focused solely on osteophyte detection using MRI. Despite the models being trained only on osteophyte labels, other correlated

features captured by MRI, which also captures soft tissues, may have influenced the predictions. The whole knee DenseNet-121 model in this study achieved a kappa of 0.64 and a ROC AUC of 0.90, which is slightly lower than Tiulpin et al.'s results. The differences may be attributed to the more focused task of osteophyte detection in this study, while Tiulpin et al.'s model considered a broader range of OA features.
Furthermore, Daneshmand et al. (2024) [26] introduced several ResNet models designed for the binary classification of osteophytes in DESS MRI scans and radiographs, also obtained from the OAI dataset, across the medial and lateral sides of both the femur and tibia. The grading system they utilized for the MRI modality is the OARSI grading system, having the same scoring range (0-3) but fewer scored subregions than the MOAKS system. The model trained on MRI volumes achieved ROC AUC scores of 0.87, 0.83, 0.78, and 0.84 for the medial femur, lateral femur, medial tibia, and lateral tibia, respectively. The balanced accuracy scored values of 0.80, 0.77, 0.71, and 0.76 for the same regions, respectively. Their dataset balance was relatively similar to this study's dataset for the tibia's subregions. In comparison, the models in this study achieved similar balanced accuracy values, with 0.93 in the femur medial and 0.80 in the tibia medial, demonstrating comparable performance to Daneshmand et al.'s results.

In Experiment 1, a binary classification model was developed to detect osteophytes in resampled whole knee MRI images, focusing on the presence of osteophytes of MOAKS grade 2 or higher. The initial step involved comparing the performance of ResNet-50 and DenseNet-121 models, both trained for 30 epochs, to identify the better-performing model for further training. The results indicated that DenseNet-121 outperformed ResNet-50, leading to its selection for extended training up to 60 epochs. Noteworthy is that the performance metrics of the experimental and final DenseNet model are very similar, while the final model trained for double the amount of epochs. However, the loss function of the experimental model has an earlier convergence than the final model and starts plateauing, indicating that the model reached its optimal performance relatively quickly and additional training would not significantly improve its performance. The final model, however, shows a more gradual and continuous decrease in loss, suggesting that extended training allowed it to refine its parameters further, albeit with diminishing returns. This behavior highlights the robustness of the DenseNet-121 architecture, capable of achieving near-optimal performance within a relatively short number of epochs, and suggests that while extended training can fine-tune the model, it does not necessarily lead to substantial performance gains once the model has converged. Furthermore, since these models were trained on whole knee images, various OA-related features could be visible on the MRI images. The progression of certain OA-related features, such as cartilage degradation, subchondral bone sclerosis, and synovitis, often occur concomitantly with osteophyte formation and are interrelated [27]. Consequently, these models might unintentionally learn to detect features associated not just with osteophytes but also with other OA-related changes, particularly in the higher grades of osteophyte scores. Given that this is a binary classification model with a threshold set at >1, the filters detecting these additional features could negatively influence performance. The model might misinterpret or overemphasize these co-occurring features, leading to incorrect classifications and reduced specificity in detecting osteophytes alone. This unintended learning could introduce noise into the decision-making process, thereby impacting the overall accuracy and reliability of the model, resulting in lower maximally achievable performance metrics.

Building on the setup from Experiment 1, Experiment 2 involved training six models to detect osteophytes with a MOAKS grade of 1 or higher in six predefined ROIs. Each model was trained for 150 epochs to optimize performance in detecting osteophytes in these specific regions. While this approach allows for osteophyte detection in specific regions, the models' performances vary across different ROIs, due to class imbalances. Future research could involve balancing the training data across all ROIs to ensure consistency and improve overall performances or at least provide equal training grounds, for inter-subregion comparisons. Expanding the analysis to include more or all MOAKS subregions could provide a more comprehensive assessment of OA severity, though this extension would require handling increased computational complexity, and more detailed annotations and segmentation ability. Furthermore, a weighted loss function could enhance the performance of subregions with poorly balanced classes, as only the superior and inferior sides of the patella had relatively balanced classes.

In Experiment 3, the focus was on the sub-region with the most balanced class ratio, the lateral side of the femur, to train a multi-class classification model. Various loss functions, including CCE, WCCE, and focal loss, were tested to identify the most effective for improving model performance. Additionally, learning rates of 0.01 and 0.0001 were compared to the standard learning rate, and models were also trained on resampled versions of cropped input images to evaluate performance and training time. While focusing on the most balanced sub-region provided valuable insights, the results may not generalize well to other sub-regions with less balanced class distributions.

Future work should include a broader range of sub-regions to enhance generalizability. Further experimentation with other advanced loss functions tailored for ordinal data, given the ordinal nature of MOAKS grading, could yield better performance. Examples of this would be the novel ordinal loss proposed by Chen et al. (2019) [28], developed specifically for grading OA, or an ordinal cross-entropy loss, which works similarly to the CCE loss, but penalizes misclassifications based on class proximity. This study explored learning rates in powers of ten; more granular tuning of the learning rate, potentially using adaptive learning rate schedules, could optimize the training process further. Implementing early stopping based on validation loss and increasing dropout rates could prevent overfitting and improve generalization capabilities.

In Experiment 4, significant differences in performance metrics were observed between the results from previous experiments on the validation set and the external test set from the EMC, with the validation set performing better. This discrepancy can be attributed to the differences in imaging modalities; the validation set comprised DESS images, while the external test set used FSPGR-FS images. DESS images are known to provide higher quality and more detailed knee joint structures, due to better evaluation of cartilage and thus a better contrast signal relative to bone [29], which likely contributed to better model performance. However, since a large number of imaging sequences are available for mapping osteophytes [3], for future research, osteophyte detection and grading DL models should be trained on a dataset with a mix of imaging sequences to enhance robustness across different imaging modalities. This approach would help ensure that the models perform consistently well regardless of the specific imaging sequence used, making them more applicable in diverse clinical settings.

Furthermore, the younger demographic in the external test set likely influenced the models' performances. As individuals age, the knee morphology undergoes natural changes, even in the absence of OA, which could impact how well the models generalize to different age groups [30]. Incorporating a broader age range in the training set is an important step for future research to ensure that the models can perform consistently across various demographic groups. This broader representation could lead to improved accuracy and robustness when applied to younger or older populations.

Additionally, the external test set exhibited severe class imbalance, as seen in Figure 4, compared to the training/validation sets shown in Figure 1. This imbalance, particularly with a high prevalence of grade 1 osteophytes, posed a challenge for the models, as positive cases being grade 1 are often the hardest to classify accurately based on the multi-class models' metrics and confusion matrices presented in this study. As can be seen from the confusion matrices, the binary models often had a harder time grading the positive class correctly than grading the negative class correctly, with more FN predictions than FP predictions in most cases. This indicates that the models were more likely to miss detecting osteophytes when they were present. This made it even harder for models operating on unbalanced external test sets to achieve similar results to the validation metric results.

In all experiments, there is a potential for bias in the validation results due to the inclusion of images from different time points of the same patient in both the training and validation sets. This could lead the model to memorize features from the same knee, especially if the orientation and placement during imaging were consistent across sessions. However, several factors mitigate this risk. Changes in the knee morphology, such as the development or progression of osteophytes or other abnormalities, introduce variability that makes it harder for the model to simply recall earlier images. Additionally, variations in imaging conditions, such as positioning or slight differences in anatomy presentation, further reduce the chances of memorization. The use of an external test set ensures that the model's true performance is evaluated on completely unseen data, eliminating overlap concerns there. While this issue is worth mentioning, its impact on the study's overall findings is likely limited.

While this study has laid a solid foundation, several areas for further research can be pursued to enhance the findings and applicability. Incorporating gradient-weighted Class Activation Mapping (Grad-CAM) for visual explanations of the model's decision-making process can significantly improve interpretability and trustworthiness. According to Adebayo et al. (2020) [31], methods like Grad-CAM generally pass sanity checks, making them more reliable compared to saliency maps, which can be noisy and less precise. Layer-wise relevance propagation, while detailed, is more complex and computationally intensive. Grad-CAM, offering a balance of interpretability and computational efficiency, produces intuitive heatmaps that highlight important regions in input images, making it ideal for medical imaging tasks. This approach helps understand which parts of the MRI images the model focuses on, enhancing the model's transparency.

Systematic hyperparameter tuning using techniques such as grid search or Bayesian optimization could lead to better model performance and more robust results. Grid search involves exhaustively searching through a specified subset of the hyperparameter space, which is effective but computationally expensive. In contrast, Bayesian optimization builds a probabilistic model of

the objective function and uses it to select the most promising hyperparameters, making it more efficient. Bayesian optimization considers past evaluations to make informed decisions, often outperforming grid and random search methods in terms of both search time and model performance [32]. Bayesian optimization significantly enhanced CNN accuracy in brain tumor classification from MRI scans in the study of Amou et al. (2022) [33] and could potentially improve the accuracy and robustness of the proposed models in this study. This study did not focus on the optimization of hyperparameters, which presents an opportunity for future research to implement these techniques and potentially enhance the model's performance and robustness.
Furthermore, increasing the external test set's size while maintaining the natural class distribution, rather than artificially balancing the external test set, would provide a more accurate evaluation of the model's performance and generalizability across all classes.

Investigating other network architectures beyond DenseNet-121 and ResNet-50 could potentially offer improvements in accuracy and efficiency. Including more MOAKS subregions in the analysis could provide a more comprehensive assessment of OA severity. This would require extending the current framework to handle a greater variety of subregions and potentially dealing with increased computational complexity.

To optimize the training process, implementing early stopping during model training based on validation loss could significantly prevent overfitting and enhance the model's generalization capabilities. Early stopping is a technique where training is halted once the model's performance on a validation set stops improving, effectively allowing the model to train until it cannot get any better. This prevents the model from learning noise in the training data, which often leads to overfitting, thereby ensuring that it generalizes well to new, unseen data. In essence, early stopping ensures that the training process is not only more efficient but also more effective in producing robust and generalizable models.
Moreover, continuous monitoring and dynamic adjustment of the training process can lead to more efficient training and better model performance. For instance, early stopping can be combined with adaptive learning rate schedules, where the learning rate is reduced when the model's performance plateaus. This approach helps in fine-tuning the model parameters more effectively during the later stages of training, avoiding the risk of overshooting the optimal values.

For clinical implementation, it is essential to address practical aspects such as the integration of the DL models into existing medical workflows, ensuring the models are user-friendly and transparent for radiologists, and validating the models in diverse clinical settings to confirm their effectiveness and reliability.

By addressing these areas, future research can build upon the foundations laid in this study to develop more accurate, interpretable, and clinically useful models for osteophyte assessment.

6 Conclusion

This study has made progress in the automatic detection and grading of osteophytes on MRI volumes using DL techniques, specifically utilizing the DenseNet-121 and ResNet-50 architectures. The findings demonstrated that DenseNet-121 generally outperforms ResNet50, achieving early convergence and optimal performance within fewer epochs.
This study revealed that a localized analysis of specific ROIs can enhance detection accuracy, although imbalanced datasets within these subregions remain challenging. Additionally, while the WCCE loss function improved multi-class classification, further optimization of hyperparameters and comparisons with other established models in the medical computer vision sector is necessary for clinical implementation.
The external validation highlighted a significant drop in performance due to variations in imaging modalities and patient demographics, underscoring the models' sensitivity to diverse data characteristics. This suggests a need for more robust models capable of maintaining high performance across different datasets and clinical settings.
Future work should focus on optimization techniques like Bayesian hyperparameter tuning and training on mixed imaging sequences to improve robustness and generalizability. Comparative studies with established models in medical computer vision, along with integrating Grad-CAM for interpretability, are vital. Handling imbalanced data using weighted loss functions and data augmentation and experimenting with ordinal loss functions tailored to the MOAKS grading system will further enhance the models' performance. Implementing further regularization methods like dropout and early stopping, combined with expanding the dataset to include a wider variety of patient demographics and imaging sequences, will enhance and validate the robustness of the model. These measures will advance the development of a clinically viable model for the automatic detection and grading of osteophytes.

References

[1] Nigel Arden and Michael C. Nevitt. "Osteoarthritis: epidemiology". eng. In: *Best Practice & Research. Clinical Rheumatology* 20.1 (Feb. 2006), pp. 3–25. ISSN: 1521-6942. DOI: 10.1016/j.berh.2005.09.007.

[2] Anita E. Wluka, Cate B. Lombard, and Flavia M. Cicuttini. "Tackling obesity in knee osteoarthritis". eng. In: *Nature Reviews. Rheumatology* 9.4 (Apr. 2013), pp. 225–235. ISSN: 1759-4804. DOI: 10.1038/nrrheum.2012.224.

[3] D.J. Hunter et al. "Evolution of semi-quantitative whole joint assessment of knee OA: MOAKS (MRI Osteoarthritis Knee Score)". en. In: *Osteoarthritis and Cartilage* 19.8 (Aug. 2011), pp. 990–1002. ISSN: 10634584. DOI: 10.1016/j.joca.2011.05.004. URL: https://linkinghub.elsevier.com/retrieve/pii/S1063458411001531 (visited on 12/05/2023).

[4] R. D. Altman et al. "Atlas of individual radiographic features in osteoarthritis". eng. In: *Osteoarthritis and Cartilage* 3 Suppl A (Sept. 1995), pp. 3–70. ISSN: 1063-4584.

[5] Samuel Newman, Huzefah Ahmed, and Nader Rehmatullah. "Radiographic vs. MRI vs. arthroscopic assessment and grading of knee osteoarthritis - are we using appropriate imaging?" In: *Journal of Experimental Orthopaedics* 9.1 (Jan. 2022), p. 2. ISSN: 2197-1153. DOI: 10.1186/s40634-021-00442-y. URL: https://doi.org/10.1186/s40634-021-00442-y (visited on 06/11/2024).

[6] Edwin H.G. Oei et al. "3D MRI in Osteoarthritis". en. In: *Seminars in Musculoskeletal Radiology* 25.03 (June 2021), pp. 468–479. ISSN: 1089-7860, 1098-898X. DOI: 10.1055/s-0041-1730911. URL: http://www.thieme-connect.de/DOI/DOI?10.1055/s-0041-1730911 (visited on 02/06/2024).

[7] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". en. In: *Nature* 521.7553 (May 2015). Publisher: Nature Publishing Group, pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539. URL: https://www.nature.com/articles/nature14539 (visited on 07/15/2024).

[8] Richard Kijowski, Jan Fritz, and Cem M. Deniz. "Deep learning applications in osteoarthritis imaging". en. In: *Skeletal Radiology* 52.11 (Nov. 2023), pp. 2225–2238. ISSN: 1432-2161. DOI: 10.1007/s00256-023-04296-6. URL: https://doi.org/10.1007/s00256-023-04296-6 (visited on 12/13/2023).

[9] Jean-Baptiste Schiratti et al. "A deep learning method for predicting knee osteoarthritis radiographic progression from MRI". In: *Arthritis Research & Therapy* 23.1 (Oct. 2021), p. 262. ISSN: 1478-6362. DOI: 10.1186/s13075-021-02634-4. URL: https://doi.org/10.1186/s13075-021-02634-4 (visited on 06/11/2024).

[10] C. G. Peterfy, E. Schneider, and M. Nevitt. "The osteoarthritis initiative: report on the design rationale for the magnetic resonance imaging protocol for the knee". In: *Osteoarthritis and Cartilage* 16.12 (Dec. 2008), pp. 1433–1441. ISSN: 1063-4584. DOI: 10.1016/j.joca.2008.06.016. URL: https://www.sciencedirect.com/science/article/pii/S1063458408002239 (visited on 08/26/2024).

[11] David J. Hunter et al. "Biomarkers for osteoarthritis: Current position and steps towards further validation". In: *Best Practice & Research Clinical Rheumatology*. Osteoarthritis: Moving from Evidence to Practice 28.1 (Feb. 2014), pp. 61–71. ISSN: 1521-6942. DOI: 10.1016/j.berh.2014.01.007. URL: https://www.sciencedirect.com/science/article/pii/S1521694214000084 (visited on 08/26/2024).

[12] Xiaoyu Li, Chunpu Li, and Peng Zhang. "Predictive models of radiographic progression and pain progression in patients with knee osteoarthritis: data from the FNIH OA biomarkers consortium project". In: *Arthritis Research & Therapy* 26.1 (May 2024), p. 112. ISSN: 1478-6362. DOI: 10.1186/s13075-024-03346-1. URL: https://doi.org/10.1186/s13075-024-03346-1 (visited on 08/23/2024).

[13] *The osteoarthritis initiative: report on the design rationale for the magnetic resonance imaging protocol for the knee - ScienceDirect*. URL: https://www.sciencedirect.com/science/article/pii/S1063458408002239?via%3Dihub (visited on 08/26/2024).

[14] Frank W. Roemer et al. "Semi-quantitative MRI biomarkers of knee osteoarthritis progression in the FNIH biomarkers consortium cohortMethodologic aspects and definition of change". In: *BMC Musculoskeletal Disorders* 17.1 (Nov. 2016), p. 466. ISSN: 1471-2474. DOI: 10.1186/s12891-016-1310-6. URL: https://doi.org/10.1186/s12891-016-1310-6 (visited on 08/26/2024).

[15] Rianne A. van der Heijden et al. "Structural Abnormalities on Magnetic Resonance Imaging in Patients With Patellofemoral Pain: A Cross-sectional Case-Control Study". eng. In: *The American Journal of Sports Medicine* 44.9 (Sept. 2016), pp. 2339–2346. ISSN: 1552-3365. DOI: 10.1177/0363546516646107.

[16] Xin Yang et al. "Efficacy of magnetic resonance imaging with an SPGR sequence for the early evaluation of knee cartilage degeneration and the relationship between cartilage and other tissues". In: *Journal of Orthopaedic Surgery and Research* 14.1 (May 2019), p. 152. ISSN: 1749-799X. DOI: 10.1186/s13018-019-1172-3. URL: https://doi.org/10.1186/s13018-019-1172-3 (visited on 08/29/2024).

[17] Maria Grazia Chiarilli et al. "Bone marrow magnetic resonance imaging: physiologic and pathologic findings that radiologist should know". en. In: *La radiologia medica* 126.2 (Feb. 2021), pp. 264–276. ISSN: 1826-6983. DOI: 10.1007/s11547-020-01239-2. URL: https://doi.org/10.1007/s11547-020-01239-2 (visited on 09/09/2024).

[18] M. Jorge Cardoso et al. *MONAI: An open-source framework for deep learning in healthcare*. en. arXiv:2211.02701 [cs]. Nov. 2022. URL: http://arxiv.org/abs/2211.02701 (visited on 09/02/2024).

[19] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. arXiv:1412.6980 [cs]. Jan. 2017. DOI: 10.48550/arXiv.1412.6980. URL: http://arxiv.org/abs/1412.6980 (visited on 07/18/2024).

[20] Ines R. Campos et al. *Prediction of Knee Osteoarthritis using MRI-based Radiomic Features of the Subchondral Bone and Infrapatellar Fat Pad*. meeting abstract. 2024.

[21] Gao Huang et al. *Densely Connected Convolutional Networks*. arXiv:1608.06993 [cs]. Jan. 2018. DOI: 10.48550/arXiv.1608.06993. URL: http://arxiv.org/abs/1608.06993 (visited on 06/27/2024).

[22] Binge Cui, Xin Chen, and Yan Lu. "Semantic Segmentation of Remote Sensing Images Using Transfer Learning and Deep Convolutional Neural Network With Dense Connection". In: *IEEE Access* PP (June 2020), pp. 1–1. DOI: 10.1109/ACCESS.2020.3003914.

[23] Kaiming He et al. *Deep Residual Learning for Image Recognition*. arXiv:1512.03385 [cs]. Dec. 2015. DOI: 10.48550/arXiv.1512.03385. URL: http://arxiv.org/abs/1512.03385 (visited on 06/27/2024).

[24] Tsung-Yi Lin et al. *Focal Loss for Dense Object Detection*. arXiv:1708.02002 [cs] version: 2. Feb. 2018. URL: http://arxiv.org/abs/1708.02002 (visited on 02/21/2024).

[25] Aleksei Tiulpin et al. "Automatic Knee Osteoarthritis Diagnosis from Plain Radiographs: A Deep Learning-Based Approach". In: *Scientific Reports* 8 (Jan. 2018), p. 1727. ISSN: 2045-2322. DOI: 10.1038/s41598-018-20132-7. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5789045/ (visited on 12/05/2023).

[26] Mitra Daneshmand et al. "Deep learning based detection of osteophytes in radiographs and magnetic resonance imagings of the knee using 2D and 3D morphology". en. In: *Journal of Orthopaedic Research* 42.7 (2024). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jor.25800, pp. 1473–1481. ISSN: 1554-527X. DOI: 10.1002/jor.25800. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/jor.25800 (visited on 09/06/2024).

[27] Guangyi Li et al. "Subchondral bone in osteoarthritis: insight into risk factors and microstructural changes". In: *Arthritis Research & Therapy* 15.6 (2013), p. 223. ISSN: 1478-6354. DOI: 10.1186/ar4405. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4061721/ (visited on 07/18/2024).

[28] Pingjun Chen et al. "Fully automatic knee osteoarthritis severity grading using deep neural networks with a novel ordinal loss". eng. In: *Computerized Medical Imaging and Graphics: The Official Journal of the Computerized Medical Imaging Society* 75 (July 2019), pp. 84–92. ISSN: 1879-0771. DOI: 10.1016/j.compmedimag.2019.06.002.

[29] *DESS MRI sequence | FADE MRI| double-echo steady-state sequence physics and image appearance*. en-US. URL: https://mrimaster.com/characterise-image-double-echo-steady-state/ (visited on 07/18/2024).

[30] Katherine Nguyen et al. "Shape modelling reveals age-related knee bony shape changes in asymptomatic knees". eng. In: *Journal of Orthopaedic Research: Official Publication of the Orthopaedic Research Society* (June 2024). ISSN: 1554-527X. DOI: 10.1002/jor.25923.

[31] Julius Adebayo et al. *Sanity Checks for Saliency Maps*. en. arXiv:1810.03292 [cs, stat]. Nov. 2020. URL: http://arxiv.org/abs/1810.03292 (visited on 06/12/2024).

[32] A. Helen Victoria and G. Maragatham. "Automatic tuning of hyperparameters using Bayesian optimization". en. In: *Evolving Systems* 12.1 (Mar. 2021), pp. 217–223. ISSN: 1868-6486. DOI: 10.1007/s12530-020-09345-2. URL: https://doi.org/10.1007/s12530-020-09345-2 (visited on 07/22/2024).

[33] Mohamed Ait Amou et al. "A Novel MRI Diagnosis Method for Brain Tumor Classification Based on CNN and Bayesian Optimization". en. In: *Healthcare* 10.3 (Mar. 2022). Number: 3 Publisher: Multidisciplinary Digital Publishing Institute, p. 494. ISSN: 2227-9032. DOI: 10.3390/healthcare10030494. URL: https://www.mdpi.com/2227-9032/10/3/494 (visited on 09/10/2024).

[34] Zheng Huilin et al. "A Deep Convolutional Neural Network-Based Multi-Class Image Classification for Automatic Wafer Map Failure Recognition in Semiconductor Manufacturing". In: *Applied Sciences* 11 (Oct. 2021), p. 9769. DOI: 10.3390/app11209769.

7 Appendix

ML approaches for medical imaging

Due to the nature of most medical images being reviewed by medical specialists, there is an abundance of labeled medical data that has been labeled. Due to this, supervised learning is a powerful tool in the medical imaging field. This supervised learning approach involves training a model on labeled data, with each piece of data explicitly tagged with its correct classification. The model learns to make predictions on new, unseen data by adjusting its internal parameters (weights and biases) to minimize the error between its predictions and the actual labels. These parameters are set randomly or based on enhanced convergence or performance strategies. During forward propagation, the model processes input data through its architecture and makes a prediction of the output. The loss function evaluates these predictions and provides a numerical measure of the model's performance. To optimize the model's weights and biases, the learning algorithm calculates a gradient vector, indicating how the prediction error would change with slight adjustments to each weight. This is followed by updating the parameters in the opposite direction to the gradient, leveraging the chain rule during backpropagation. This process involves computing the gradient of the error function with respect to each weight through systematic layer-by-layer multiplication of derivatives, from the output back toward the input. The objective is to refine the model's predictive accuracy on new data by harnessing the patterns learned from the training dataset, thus making it excel at accurately predicting or classifying new instances. The loss function quantifies the difference between the model's predictions and the actual data. It serves as a guide for the optimization process, where the goal is to minimize this difference, thereby improving the model's accuracy. Different tasks may require different loss functions. The detection and grading of OA fall under the binary and multi-class classification tasks and thus require corresponding loss functions.

Optimization is the process of tuning the model's weights and biases to reduce the errors indicated by the pre-selected loss function. During backpropagation, the gradients of the loss with respect to the model's parameters are computed, and an optimization algorithm is utilized to update the model's parameters. Over the years, different algorithms have been introduced, each with its own strategy. The choice among these methods depends on various factors, including the size of the dataset, the complexity of the model, and the specific challenges of the learning task, thus making optimization not just a mechanical step but a strategic choice.

In this study, the Adaptive moment estimation (Adam) optimizer has been utilized in every model. Adam is a 2014 updated version of the RMSProp optimizer. Adam computes adaptive learning rates for each parameter by estimating the first moment and the second moment of the gradients. These estimations are slightly biased relative to the real moments and to counter this, correction factors are utilized.

Adam has gained recognition for its advanced optimization capabilities by merging the momentum technique with RMSProp's adaptive learning rate mechanism. This integration allows Adam to calculate unique adaptive learning rates for every parameter, combining the benefits of the momentum technique's smoothing effects with the ability to adjust the learning rates to the needs of each parameter. As a result, Adam proves to be very efficient across a broad spectrum of DL tasks. When compared to utilizing RMSProp alone, Adam's addition of bias-correcting factors improves reliability and performance by guaranteeing the accuracy of its estimations over time. These characteristics establish Adam as a versatile and robust optimizer, often leading to quicker convergence and improved management of sparse gradients within intricate optimization scenarios [19]. The optimization formula of Adam is presented and explained below.

Adam computes adaptive learning rates for each parameter by estimating the first moment m_t and the second moment v_t of the gradients. These estimations are slightly biased relative to the real moments and to counter this, correction factors $\hat{m}_t$ and $\hat{v}_t$ are derived. These calculations are formulated as:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{12}$$

$$v_1 = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \tag{13}$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{14}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{15}$$

Here β_1 and β_2 are forgetting factors. The parameter updating algorithm becomes:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \varepsilon} \hat{m}_t \tag{16}$$

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are specifically designed to process data in multiple array forms, such as images composed of pixel intensities across various color channels. CNNs excel in handling not just 2D data for images, but also 1D data for sequences and signals, and importantly, 3D data for volumetric imagery. The architecture of a CNN model is built upon several foundational concepts that enable it to effectively process and analyze visual data.

The architecture of a CNN model is based on several foundational concepts. First, the principle of local connections is based on the observation that in many types of data, particularly images, nearby elements are more strongly related to each other than to distant elements. This means that neurons in the convolutional layers are connected only to a small, localized area of the input. This localized view allows the network to detect features such as edges, textures, and patterns at various locations in the input data. By focusing on local connections, CNNs reduce the complexity and computational load.

Second, parameter sharing, or shared weights, is a key component of the convolutional operation, used to detect the same feature across different parts of the input data. This reduces the number of parameters and enhances generalizability. Third, pooling layers are utilized to reduce the spatial dimensions of the feature maps and thus the input data for the subsequent layers. Pooling makes the representation smaller and more manageable and introduces a form of translation invariance. This process helps reduce the sensitivity of the output to minor changes and distortions, contributing to the robustness of the model.

Fourth, the deep neural network architecture of CNNs consists of multiple layers of neurons that enable the extraction of increasingly abstract features. Early layers may detect simple features such as edges and corners, while deeper layers can identify more complex features like shapes or specific objects. This hierarchical feature extraction process is crucial for complex tasks, such as medical imaging detection and classification [7]. By integrating these four concepts, CNNs are able to achieve remarkable performances in computer vision tasks. For information on specific CNN layers, see the Appendix.

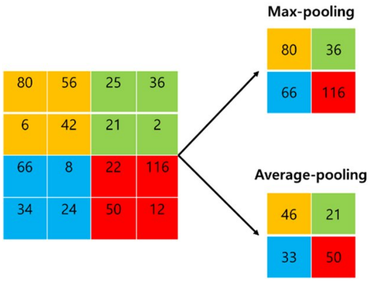
The CNN architectures are structured as a cascade of different types of layers, each with a unique role in processing and transforming the input data. Understanding the function of each layer is vital for knowing how CNNs adapt and learn.

- **Convolutional layer**: The convolutional layer is the core layer. It performs an operation called "convolution", applying filters, also known as kernels, that the network learns to detect specific features. Each convolutional layer consists of multiple different filters, sliding across the input image to produce feature maps. During training, these filters capture spatial hierarchies of features at different layers and with the convolutional operation identify features, regardless of their location in the data. The formula used to compute feature map values is shown by:

$$(I * K)[m,n] = \sum_i \sum_j K[i,j] I[m-i, n-j] \tag{17}$$

Here I represents the input data, K the kernel, m and n the rows and columns of the resulting matrix, respectively, and i and j represent the rows and columns of the kernel, respectively. This formula is used for 2D images, but with slight adjustments can be used for 3D applications.

- **Pooling layer**: Pooling layers help reduce spatial dimensions of the feature maps. The two most common pooling strategies are maximum pooling and average pooling. These methods are shown in Figure 13.



80	56	25	36
6	42	21	2
66	8	22	116
34	24	50	12

80	36
66	116

46	21
33	50

Figure 13: Two pooling methods [34].

- **Activation layer**: After each convolutional layer, an activation layer, or non-linear layer, follows. This layer introduces non-linearity to the system, enabling it to learn more complex patterns. The rectified linear unit (ReLU) is the most popular activation function, which converts all the negative numbers to zero [7]. Due to its simplicity, it reduces the computational load compared to other activation functions. ReLU can be expressed mathematically as:

$$f(x) = \max(0, x) \tag{18}$$

- **Fully connected layer**: Non-linear combinations of the extracted features can be learned at low computational cost with fully connected (FC) layers. Neurons in a FC layer are connected to all activations in the previous layer. The FC layer usually follows convolutional or pooling layers, which generate multidimensional feature maps as output. This FC layer input is flattened into vector form and fed into the FC layer. Each neuron in an FC layer computes the weighted sum of all its inputs, adds a bias term to learn offsets, and passes this value through an activation function.

- **Output layer**: The last layer in the model is the output layer. This layer determines what the model predicts, based on the output of all other layers. In classification tasks, the output layer has as many neurons as there are classes. For binary classification, a single neuron can suffice, with its output representing the probability of belonging to one of the classes. For multi-class classification, a softmax function is used. The softmax function ensures that the output neurons produce a probability distribution across classes by taking the raw scores, better known as logits, from the neurons and taking the exponential of each output and then dividing by the sum of all exponentials. For binary classification, a sigmoid function can be used, outputting a probability between zero and one. The formulas for the softmax and sigmoid function are:

$$s(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{19}$$

$$\sigma(y) = \frac{1}{1+e^{-y}} \tag{20}$$

Here, x_i is the logit for class i, j the amount of classes, and y the input to the sigmoid function.

- **Dropout layers**: Dropout layers are a regularization technique used to prevent overfitting and increase robustness. Overfitting occurs when a model learns the training data too well, capturing noise and details that negatively impact its performance on unseen data. Dropout addresses this issue by randomly "dropping out" a subset of neurons in the network during training, effectively preventing them from participating in forward propagation and backpropagation for a given iteration. A probability p determines the likelihood of a neuron dropping out. This p is a hyperparameter and set before training. The choice of p is usually between 0.1 and 0.5. This introduced randomness encourages the network to develop redundant pathways to ensure correct outcomes and prevents neurons from relying on particular neurons too much.

By understanding and utilizing these layers effectively, CNNs can be tailored to handle a wide range of image processing and analysis tasks, from simple feature detection to complex pattern recognition in medical imaging and other fields.

Performance metrics

Accuracy

The accuracy metric calculates the proportion of correct predictions made by the model, quantifying how often the model is right. Although this metric is very straightforward and easy to understand, it has some downsides. In the case of imbalanced datasets, it could be a misleading metric. A model could correctly predict only the over-sampled class and still receive a high accuracy, while this does not reflect its actual diagnostic ability, especially for the minority class. Moreover, the accuracy metric fails to distinguish between different kinds of errors. Within the realm of diagnosing OA, the impact of a false positive is, over time, less significant than that of a false negative. This is because a false negative may lead to a lack of necessary lifestyle changes.

An adjusted form of this metric to account for the problems it has with imbalanced datasets is the balanced accuracy. It addresses the issue by calculating the average of the recall and specificity obtained in each class. This provides a more fair measurement of the model's performance as it focuses more on the underrepresented classes.

Precision

The precision metric calculates the proportion of true positive predictions relative to the total number of positive predictions made. Precision is particularly informative when the cost of false positives is high or when the interest lies in the performance of the model on the positive class. When dealing with imbalanced datasets where positive cases are rare, precision can be a more relevant metric than accuracy, as it is not influenced by a large number of negative cases. However, relying solely on precision can be misleading because it does not account for the model's ability to correctly identify negative cases or its performance across other classes in multi-class scenarios.

Recall

The recall metric measures the proportion of actual positives that are correctly classified by the model. Just like precision, recall is a very suitable metric in imbalanced datasets, as it focuses solely on the ability of the model to detect the underrepresented class. Also, it is valuable in applications where missing a positive case can have serious consequences. A high recall rate ensures that the model catches as many true positive cases as possible. A limiting factor of recall is that it does not penalize the model for classifying false positives. Like this, the model can achieve high recall by simply predicting most cases as positive. Furthermore, a trade-off between recall and precision is often used for a more insightful metric, than just one of those two alone.

Specificity

The evaluation metric specificity, also known as true negative rate, is used to evaluate the proportion of actual negatives that are correctly identified as such by the model. This metric provides a clear measure of a model's ability to correctly identify instances that do not belong to a specific class. High specificity is critical for ensuring that patients without a condition are accurately identified. However, specificity alone can't be relied on, as it provides no information about the performance of the positive classes, or how well it distinguishes between multiple positive classes. Furthermore, in highly imbalanced datasets, where negative instances significantly outnumber positive instances, a model can achieve high specificity by predominantly predicting the majority class, without truly capturing the nuances necessary for identifying the minority class accurately.

F1-score
The F1-score is defined as the harmonic mean of precision and recall. It balances a trade-off between precision and recall, making it useful for scenarios where both false positives and false negatives are of concern. Also, by focusing on the harmonic mean, it offers more information on the model's performance on the underrepresented class. Although the F1-score is useful for binary classification tasks, especially when the cost for false predictions is similar, multi-class problems require averaging the F1-score across classes. This can be done with macro, micro, or weighted averaging, each of which has its implications and may not fully capture performance nuances across the classes. Moreover, while the F1-score can be more informative than accuracy in imbalanced datasets, it still can be influenced by severe imbalances.

Area under the receiver operating characteristic curve
The receiver operating characteristic (ROC) curve plots the sensitivity against the false positive rate, which is 1 - specificity. The area under the curve (AUC) quantifies the entire 2D area underneath the entire ROC curve from (0,0) to (1,1). A high ROC-AUC score indicates that the model performs well in distinguishing between the positive and negative classes. A score of 0.5 is seen as a threshold as it indicates random guessing. The ROC-AUC metric can be used for binary and, with some adjustments, multi-class classification tasks. It offers a brief overview of the model's ability to distinguish between classes, providing a single scalar value for easy comparisons. However, there are limitations to this metric. A high AUC might be achieved even if the model performs poorly on the underrepresented class because the metric primarily assesses the ability to rank predictions rather than the actual prediction accuracy for each class. Also, while it can be utilized on multi-class evaluation, it still requires an adaptation to calculate the averaged result. Moreover, the averaged AUC does not provide detailed insights into how well the model performs in each class, which is especially relevant in scenarios where the performance of underrepresented classes is critical.

Area under the precision-recall curve
Just like the F1-score, the precision-recall (PR) curve evaluates the precision-recall trade-off across different thresholds. The PR curve is more informative for imbalanced datasets, especially when the positive class is less frequent. Also, just like ROC-AUC, PR-AUC can be modified to be utilized for multi-class classification. Unlike the baseline score of 0.5 of the ROC-AUC metric that indicates random guessing, the PR-AUC baseline is variable, reflecting the class distribution's impact on the model evaluation. Usually, in imbalanced datasets, this score is lower than 0.5 because the denominator in the precision calculation is affected by the lower number of positive cases. In this case, a PR-AUC score significantly above the baseline indicates the model's excellent ability to classify rare positive cases. This variable baseline, however, makes the PR-AUC a hard metric to use to compare models across different datasets. Furthermore, while it can be used for multi-class classification problems, this process can conceal the performance nuances related to specific classes.

Multi-class classification metrics
The previously mentioned metrics are very commonly used in binary classification tasks. Two metrics that are useful for multi-class problems are Cohen's kappa metric and the confusion matrix. While these metrics also work well for binary problems, they are some of the few metrics that work just as well for binary and multi-class problems.
Cohen's kappa metric, also known as just the kappa metric, is used to evaluate the agreement between two different methods or raters. In the context of ML, it is used to assess the agreement between the model's prediction and the actual class. This is important as the severity of wrong predictions can vary based on the magnitude of the error. P_e reflects the likelihood that any agreement between the model's predictions and actual classes is due to chance alone. The kappa value ranges from -1 to 1, where 1 indicates perfect agreement, 0 indicates no agreement beyond chance, and negative indicates agreement less than chance. A disadvantage is that the single scalar it provides could be hard to interpret when it is based on a model with lots of classes.
The confusion matrix is a tool to visualize the performance of a classification model. Each row of the matrix represents the instances in a predicted class, while each column represents the instances in an actual class. It helps identify not just the errors but the types of errors, such as which classes are being confused with each other. Also, it is a useful metric as other metrics can be derived using its information. However, while it provides detailed per-class performance, the confusion matrix alone does not give a single metric that summarizes the overall performance of the model across all classes, making it challenging to quickly assess and compare models.

Data

Here are the binary distributions of the labels used in Experiment 1 and Experiment 4, when evaluating the cross-validated model from Experiment 1, presented. This means that knee MR images with a MOAKS grade of 0 or 1 are put into the negative class (0) and MR images with a MOAKS grade of 2 or 3 are put into the positive class (1).

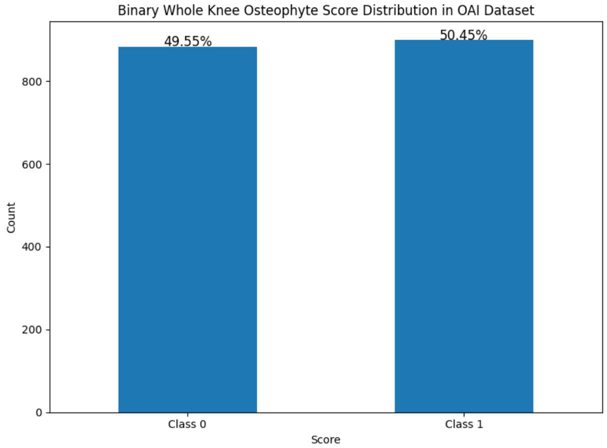


Figure 14: Distribution of the OAI dataset used for the training of the Whole Knee Resampled model.

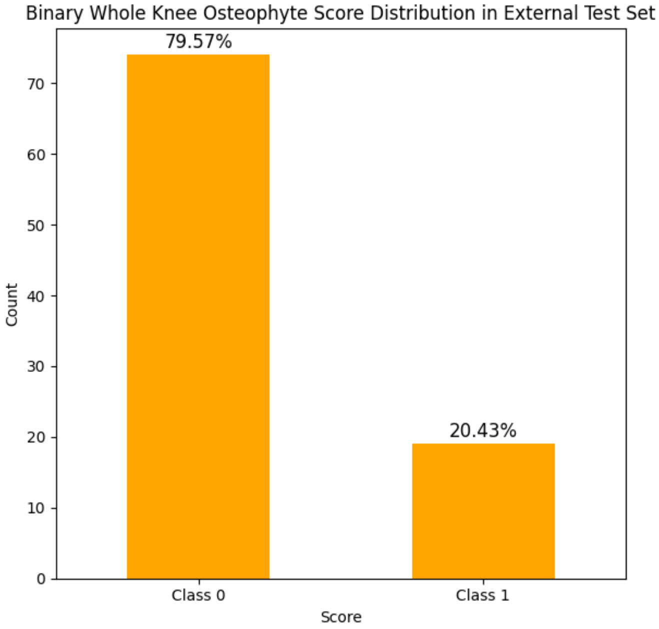
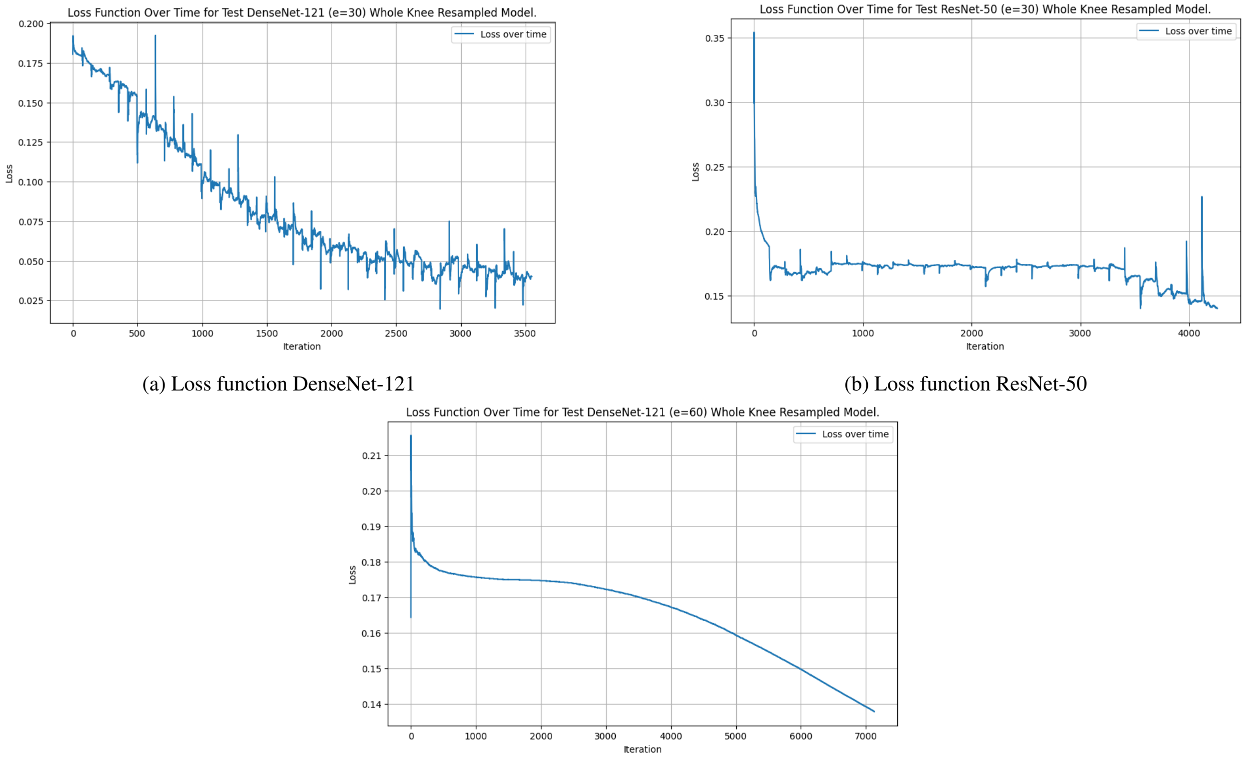


Figure 15: Distribution of the test set from the Triple P study used for the testing of the Whole Knee Resampled model.

Results

Here are the training loss functions presented for each model. Five-fold cross-validated models are depicted in five unique loss functions, one for each fold. Each iteration is the loss score, calculated once every five training batches.

Loss functions

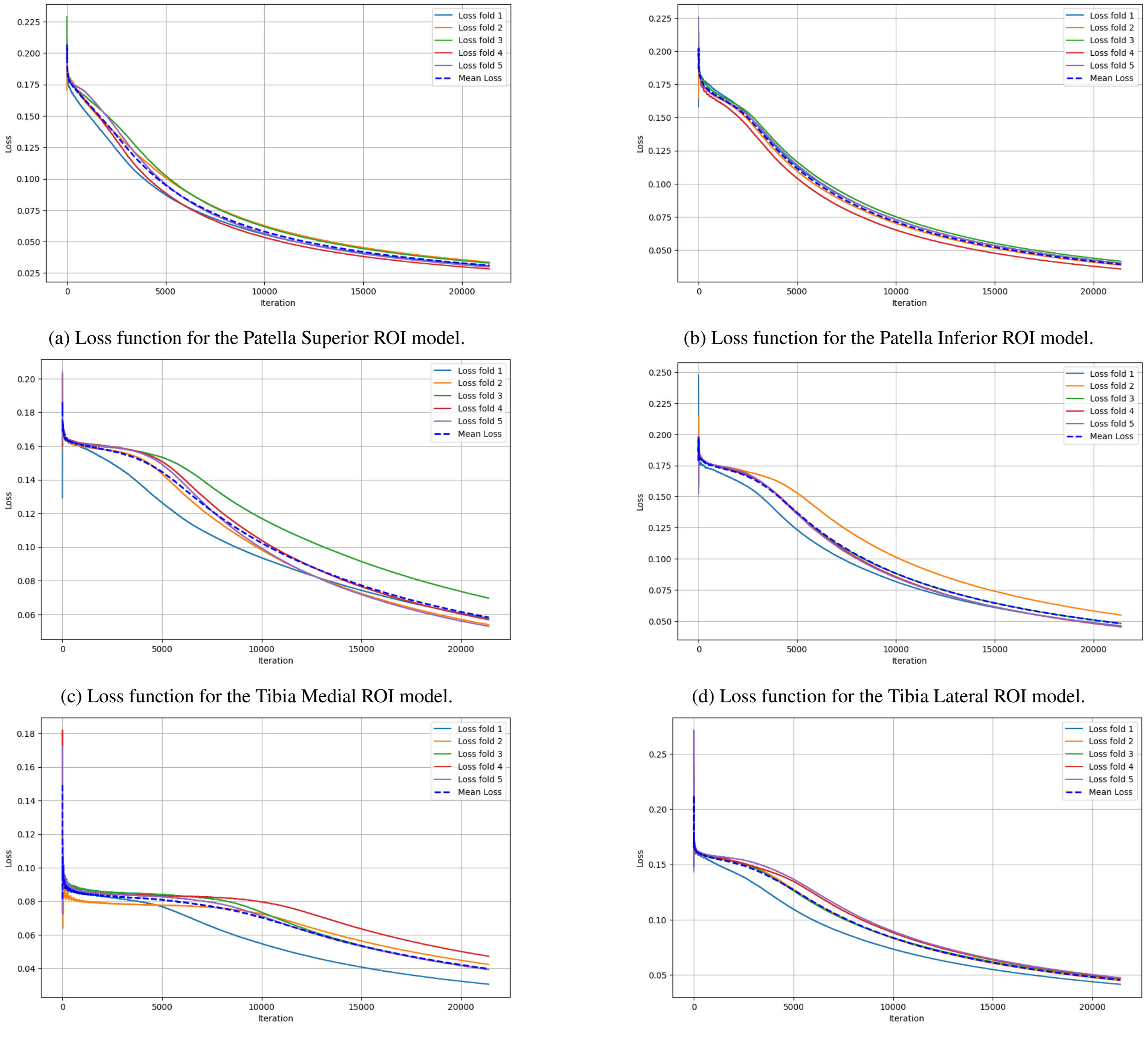


(a) Loss function DenseNet-121

(b) Loss function ResNet-50

(c) Loss function final cross-validated DenseNet-121 model

Figure 16: Loss functions over the number of iterations of Experiment 1. Each iteration depicts the calculated loss function after 5 batches.



(a) Loss function for the Patella Superior ROI model.

(b) Loss function for the Patella Inferior ROI model.

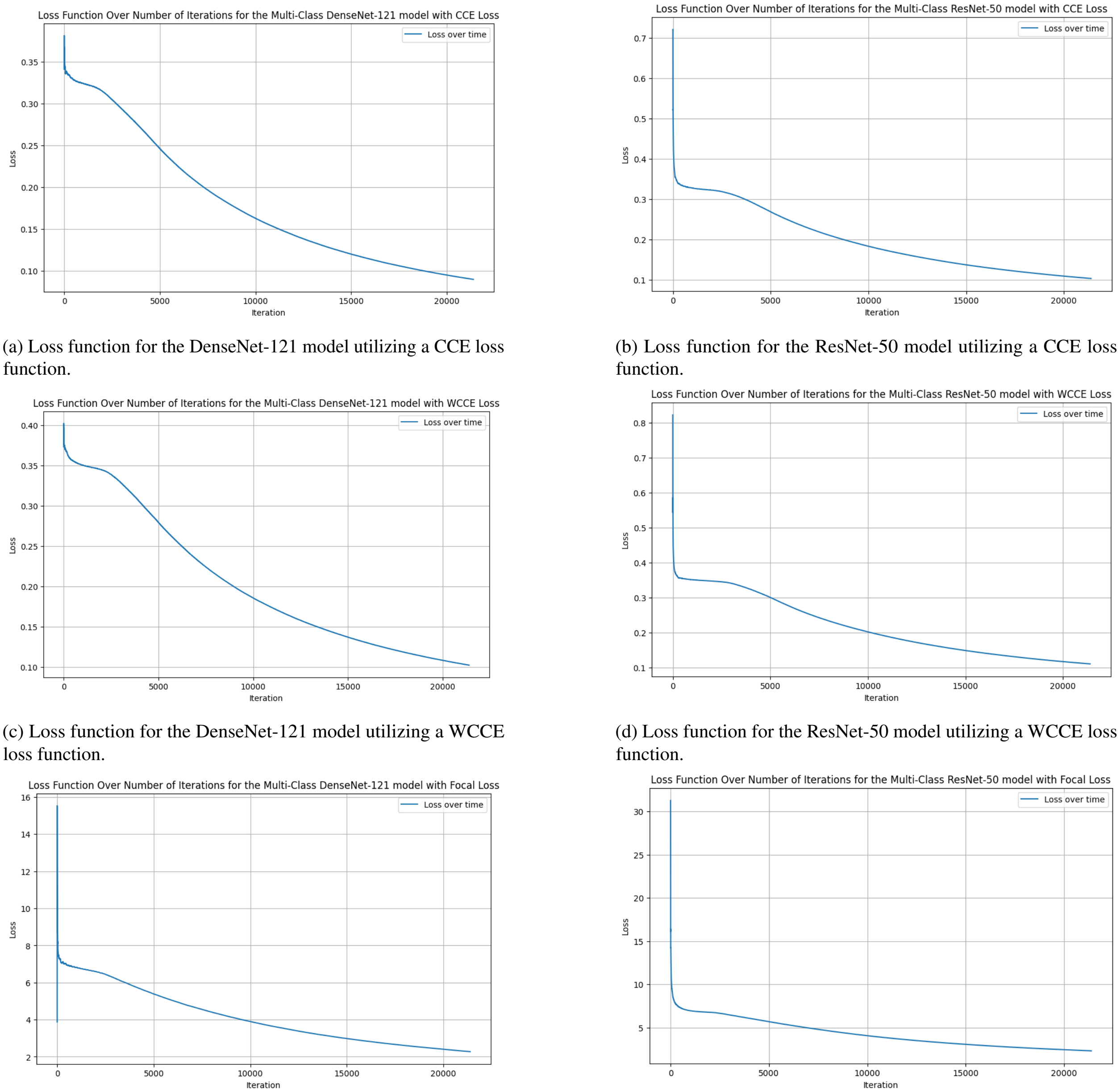
(c) Loss function for the Tibia Medial ROI model.

(d) Loss function for the Tibia Lateral ROI model.

(e) Loss function for the Femur Medial ROI model.

(f) Loss function for the Femur Lateral ROI model.

Figure 17: Loss functions over the number of iterations for the ROI DenseNet models. Each iteration depicts the calculated loss function after 5 batches.



(a) Loss function for the DenseNet-121 model utilizing a CCE loss function.

(b) Loss function for the ResNet-50 model utilizing a CCE loss function.

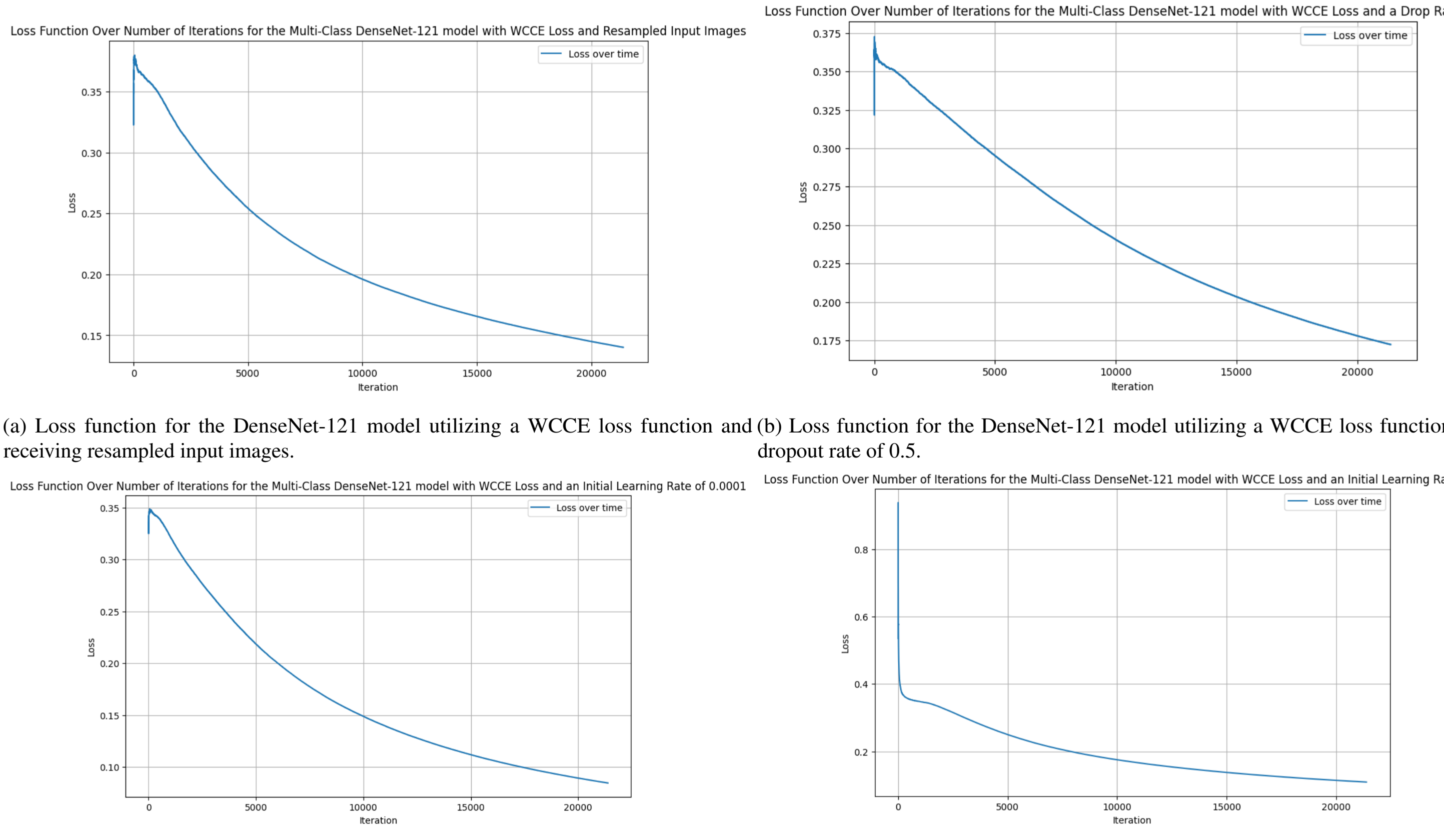
(c) Loss function for the DenseNet-121 model utilizing a WCCE loss function.

(d) Loss function for the ResNet-50 model utilizing a WCCE loss function.

(e) Loss function for the DenseNet-121 model utilizing a Focal loss function.

(f) Loss function for the ResNet-50 model utilizing a Focal loss function.

Figure 18: Loss functions over the number of iterations for the multi-class DenseNet-121 and ResNet-50 models with different loss functions. Each iteration depicts the calculated loss function after 5 batches.



(a) Loss function for the DenseNet-121 model utilizing a WCCE loss function and receiving resampled input images.

(b) Loss function for the DenseNet-121 model utilizing a WCCE loss functio dropout rate of 0.5.

(c) Loss function for the DenseNet-121 model utilizing a WCCE loss function and a learning rate of 0.0001.

(d) Loss function for the DenseNet-121 model utilizing a WCCE loss functio learning rate of 0.01

Figure 19: Loss functions over the number of iterations for the multi-class DenseNet-121 models with different hyperparameters and resampled input evaluating the femur lateral subregion. Each iteration depicts the calculated loss function after 5 batches.

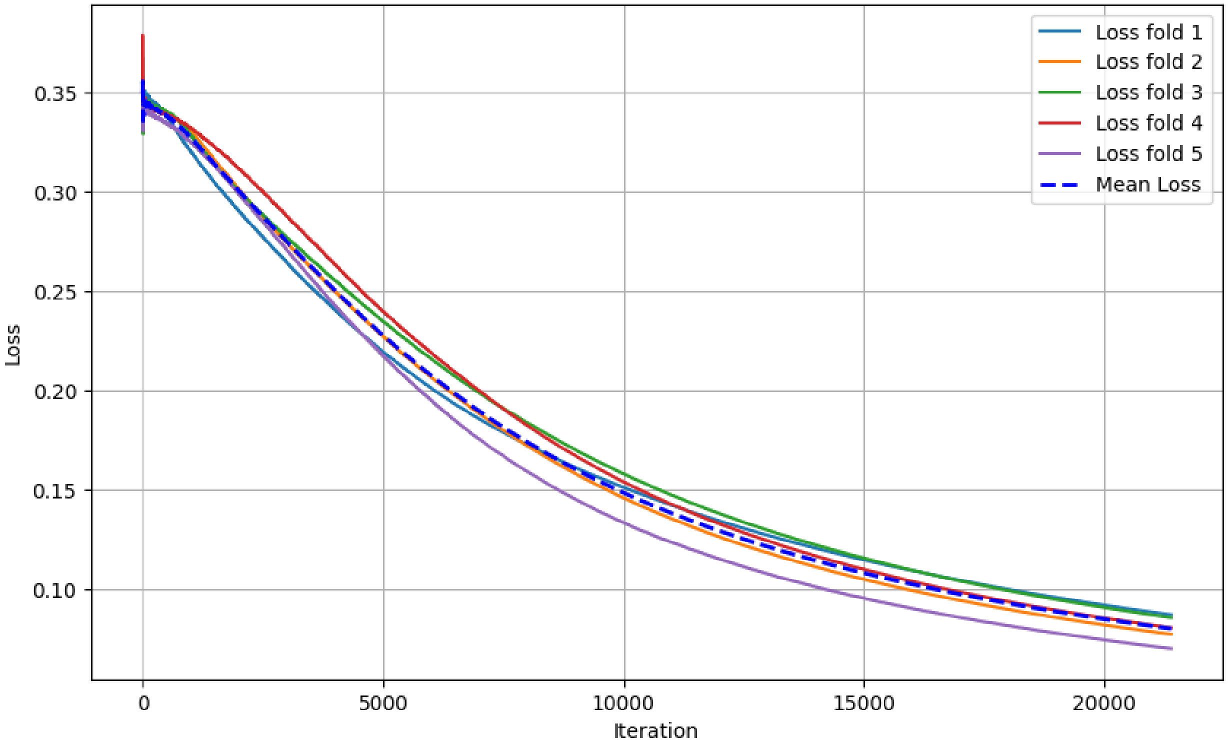
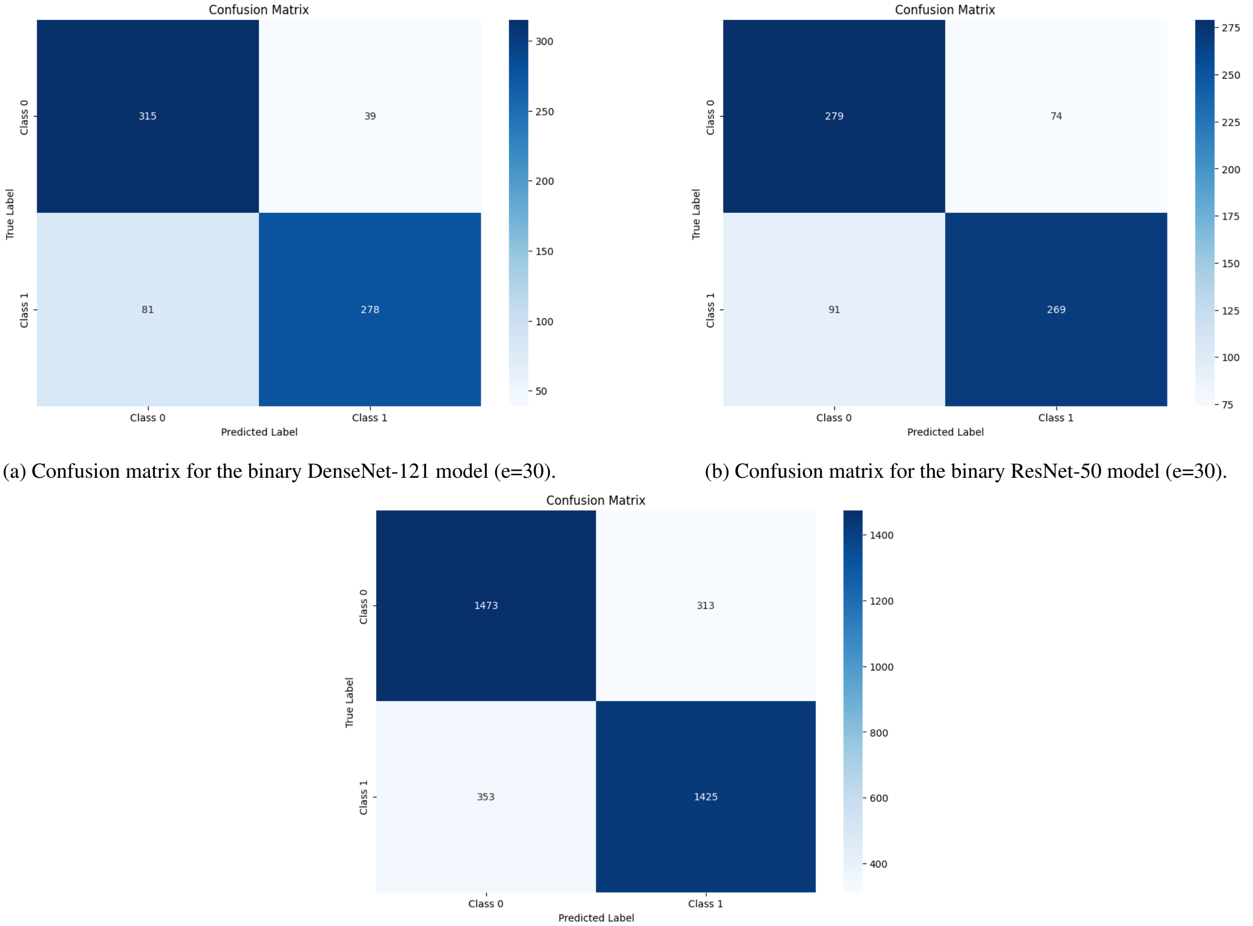


Figure 20: Loss function over iterations for the cross-validated multi-class DenseNet-121 WCCE LR = 0.0001 model trained on the femur lateral subregion.

Confusion matrices

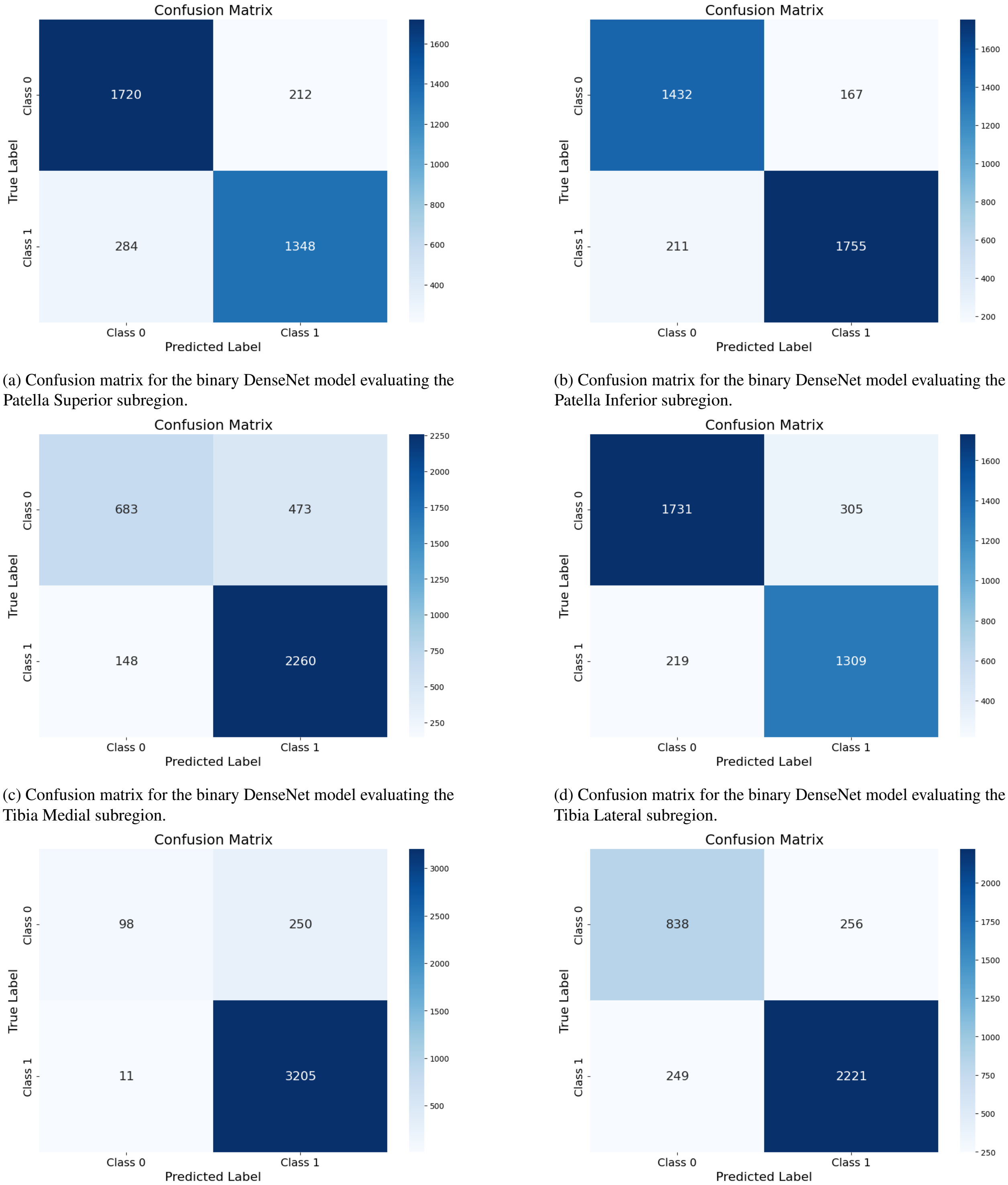


(a) Confusion matrix for the binary DenseNet-121 model (e=30).

(b) Confusion matrix for the binary ResNet-50 model (e=30).

(c) Confusion matrix for the final five-fold cross-validated binary DenseNet-121 model (e=60).

Figure 21: Confusion matrices for Experiment 1, where the whole knee images are resampled to half the voxels in every dimension. The threshold for a positive class was at a MOAKS grade of ≥ 2.



(a) Confusion matrix for the binary DenseNet model evaluating the Patella Superior subregion.

(b) Confusion matrix for the binary DenseNet model evaluating the Patella Inferior subregion.

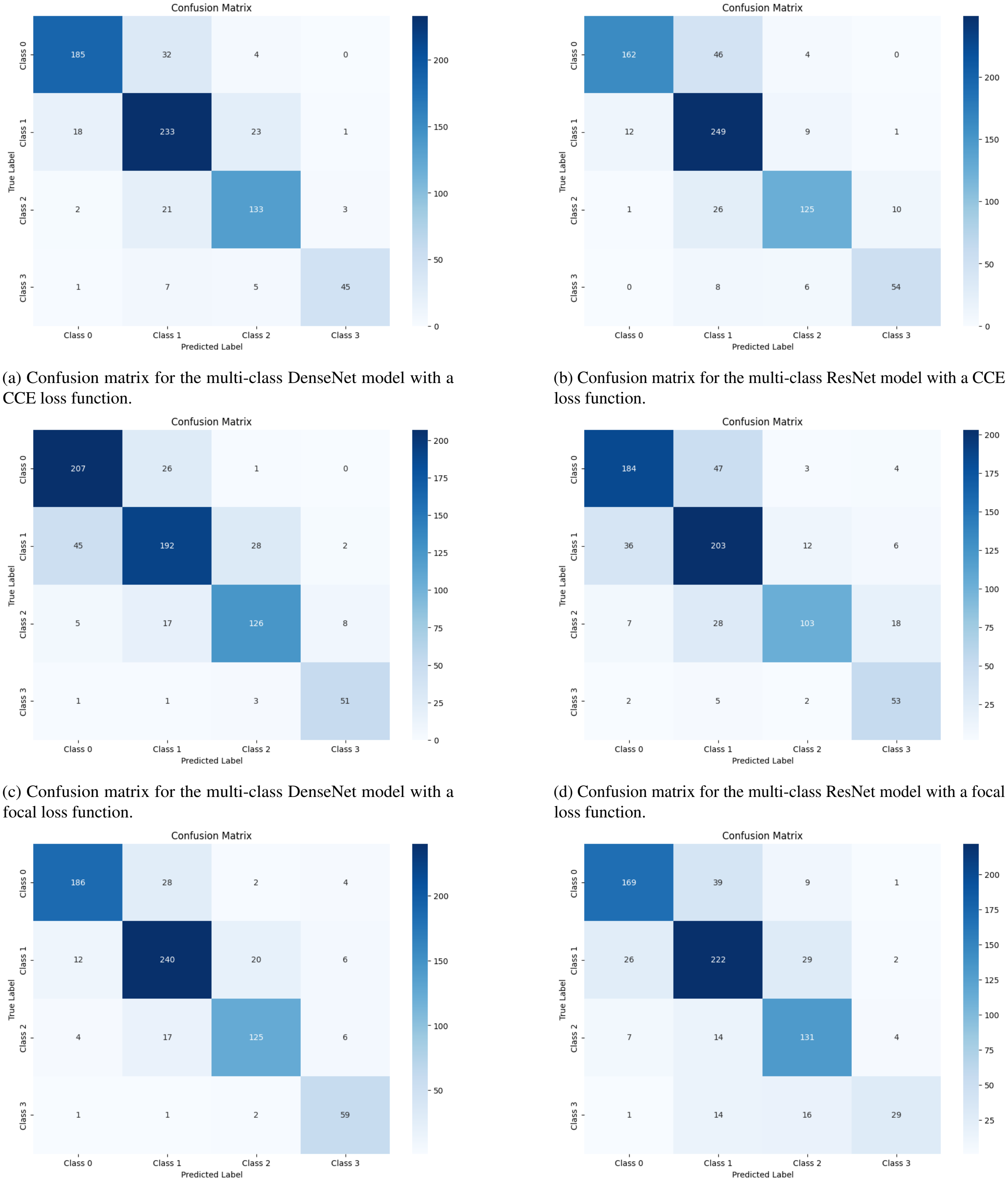
(c) Confusion matrix for the binary DenseNet model evaluating the Tibia Medial subregion.

(d) Confusion matrix for the binary DenseNet model evaluating the Tibia Lateral subregion.

(e) Confusion matrix for the binary DenseNet model evaluating the Femur Medial subregion.

(f) Confusion matrix for the binary DenseNet model evaluating the Femur Lateral subregion.

Figure 22: Confusion matrices for Experiment 2.



(a) Confusion matrix for the multi-class DenseNet model with a CCE loss function.

(b) Confusion matrix for the multi-class ResNet model with a CCE loss function.

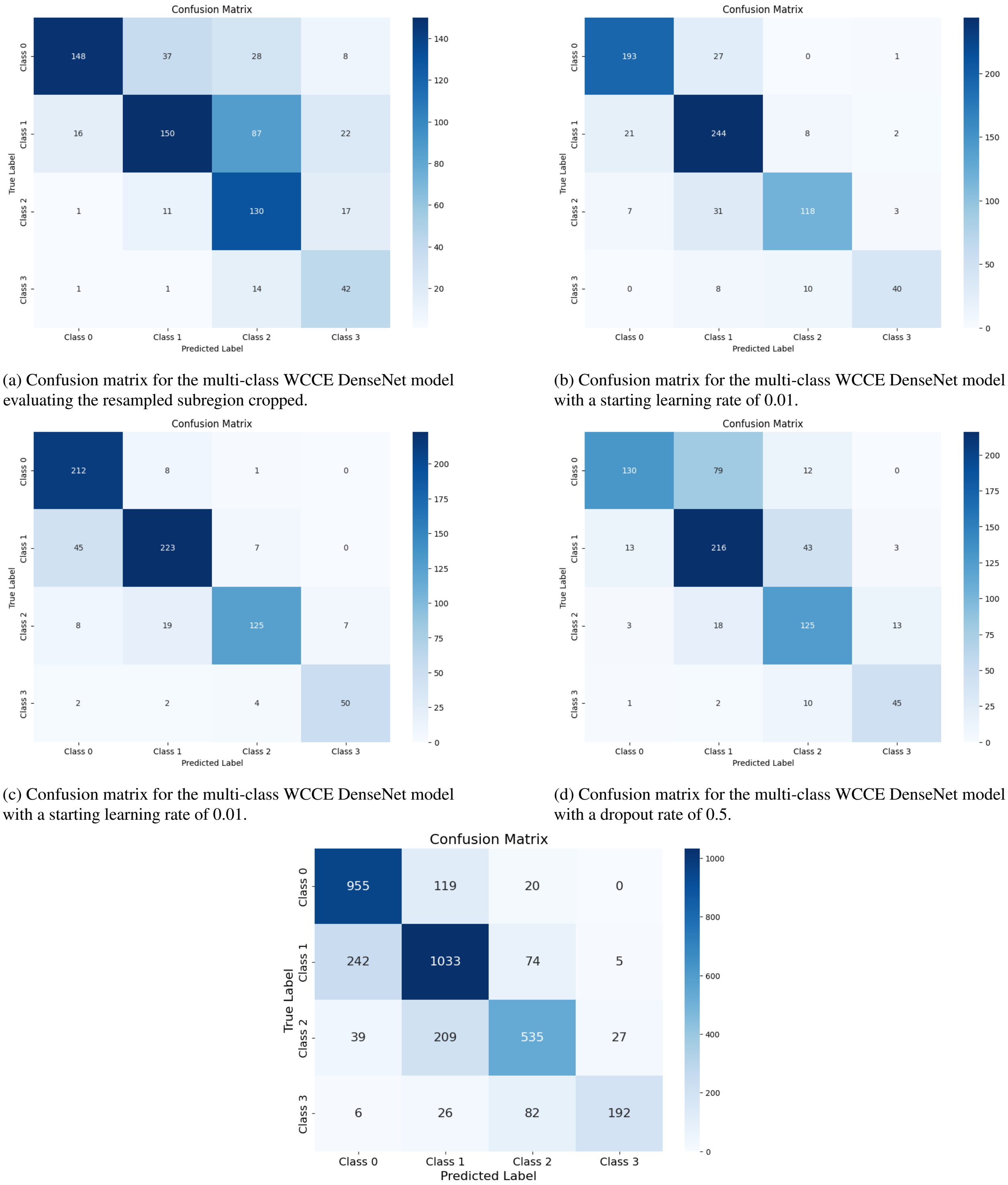
(c) Confusion matrix for the multi-class DenseNet model with a focal loss function.

(d) Confusion matrix for the multi-class ResNet model with a focal loss function.

(e) Confusion matrix for the multi-class DenseNet model with a WCCE loss function.

(f) Confusion matrix for the multi-class ResNet model with a WCCE loss function.

Figure 23: Confusion matrices for DenseNet and ResNet models with different loss functions from Experiment 3.



(a) Confusion matrix for the multi-class WCCE DenseNet model evaluating the resampled subregion cropped.

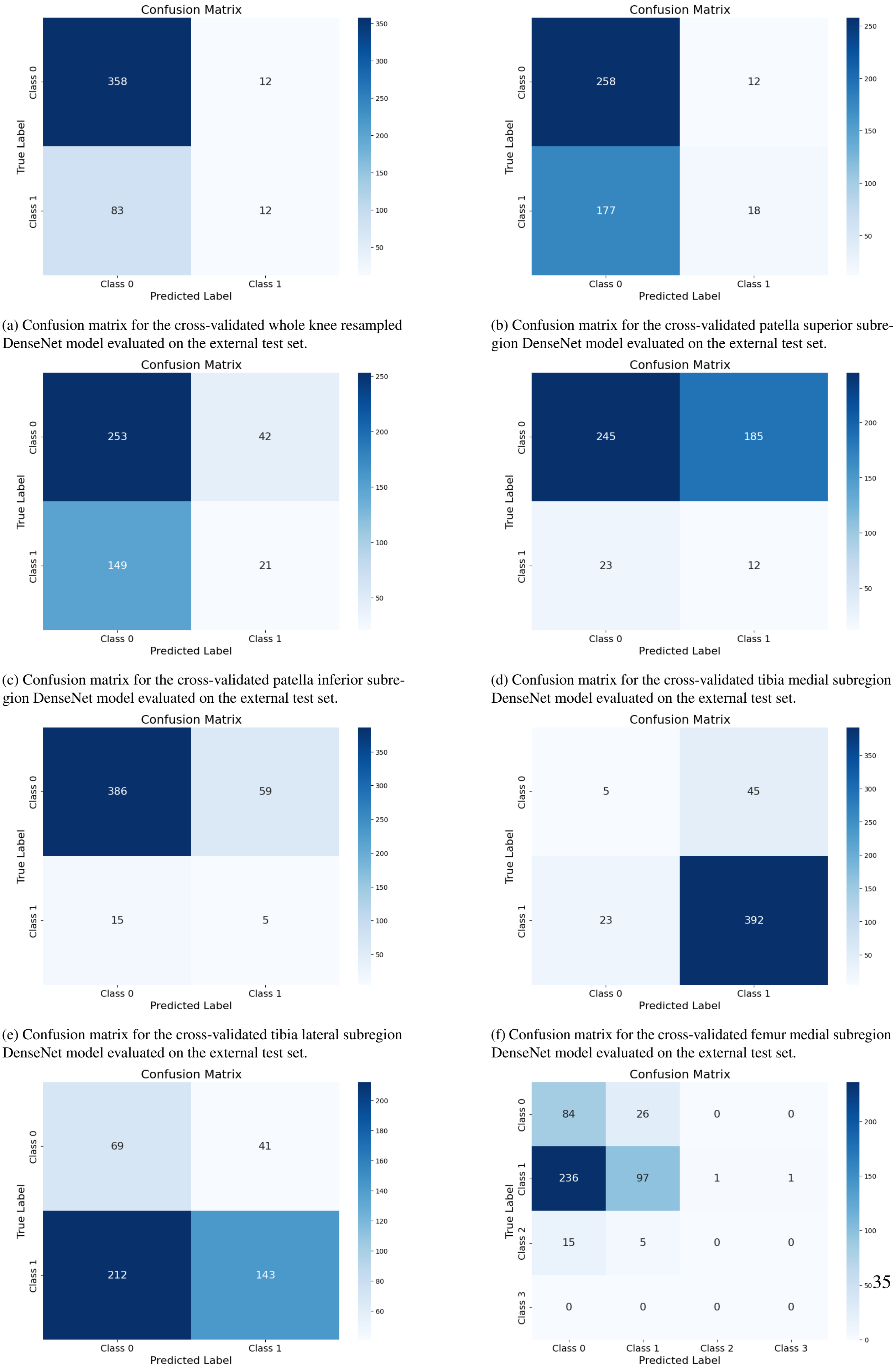
(b) Confusion matrix for the multi-class WCCE DenseNet model with a starting learning rate of 0.01.

(c) Confusion matrix for the multi-class WCCE DenseNet model with a starting learning rate of 0.01.

(d) Confusion matrix for the multi-class WCCE DenseNet model with a dropout rate of 0.5.

(e) Confusion matrix for the cross-validated multi-class WCCE DenseNet-121 with LR = 0.0001 model.

Figure 24: Confusion matrices for DenseNet models with the WCCE loss function and different hyperparameters.



(a) Confusion matrix for the cross-validated whole knee resampled DenseNet model evaluated on the external test set.

(b) Confusion matrix for the cross-validated patella superior subregion DenseNet model evaluated on the external test set.

(c) Confusion matrix for the cross-validated patella inferior subregion DenseNet model evaluated on the external test set.

(d) Confusion matrix for the cross-validated tibia medial subregion DenseNet model evaluated on the external test set.

(e) Confusion matrix for the cross-validated tibia lateral subregion DenseNet model evaluated on the external test set.

(f) Confusion matrix for the cross-validated femur medial subregion DenseNet model evaluated on the external test set.

ROC and PR AUC plots

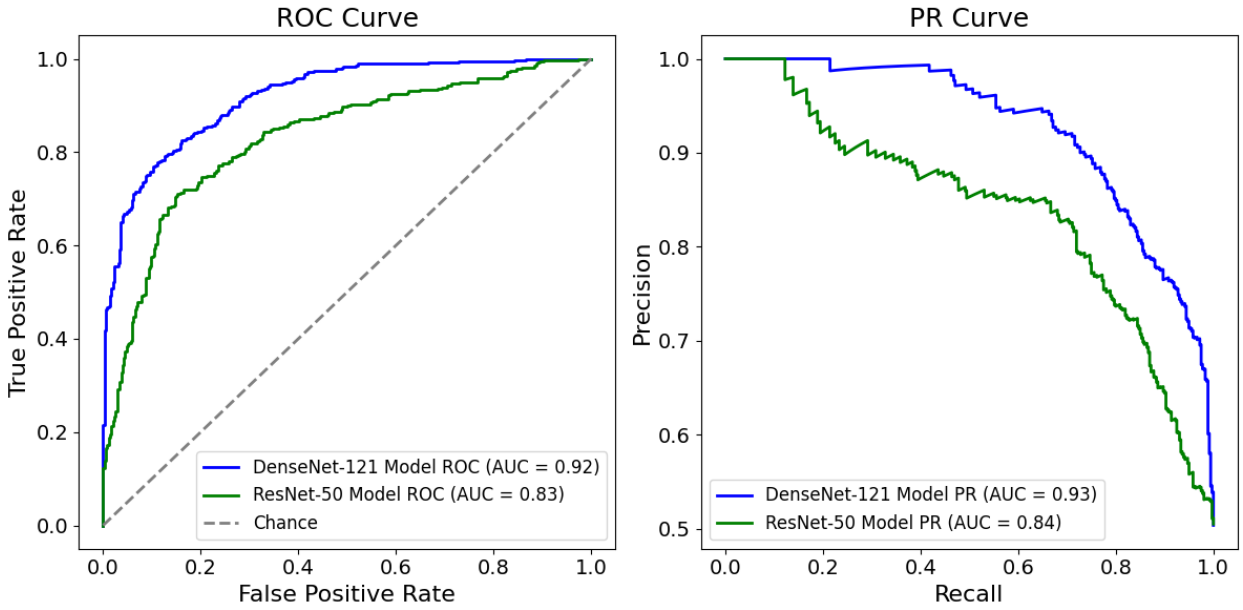


Figure 26: ROC and PR curves of the DensNet-121 and ResNet-50 models trained on the resampled whole knee images for 30 epochs each.

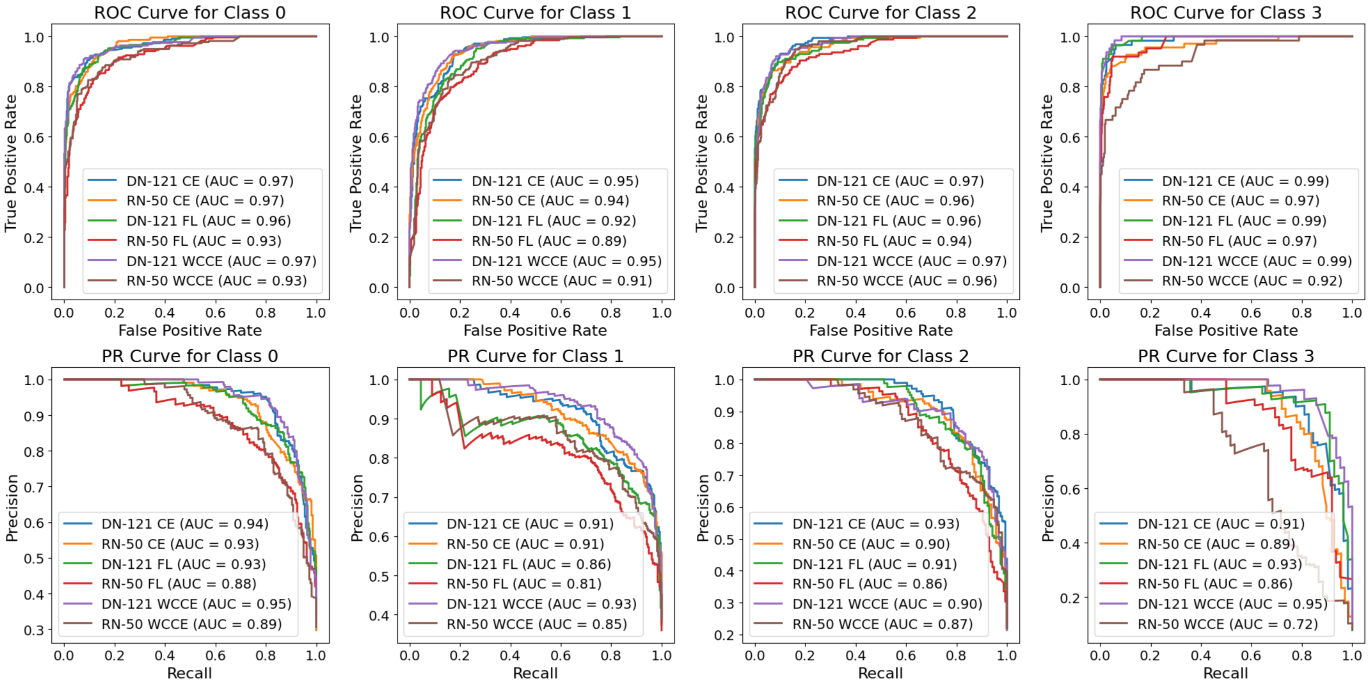


Figure 27: ROC and PR curves for all multi-class femur lateral side models with different loss functions. The ROC curve (top) and PR curve (bottom) display the performance of all 6 DenseNet-121 and ResNet-50 models, per class. Each subplot represents one class, showing how well the models distinguish between the given class and the rest. The class distribution for this subregion can be found in Figure 1.

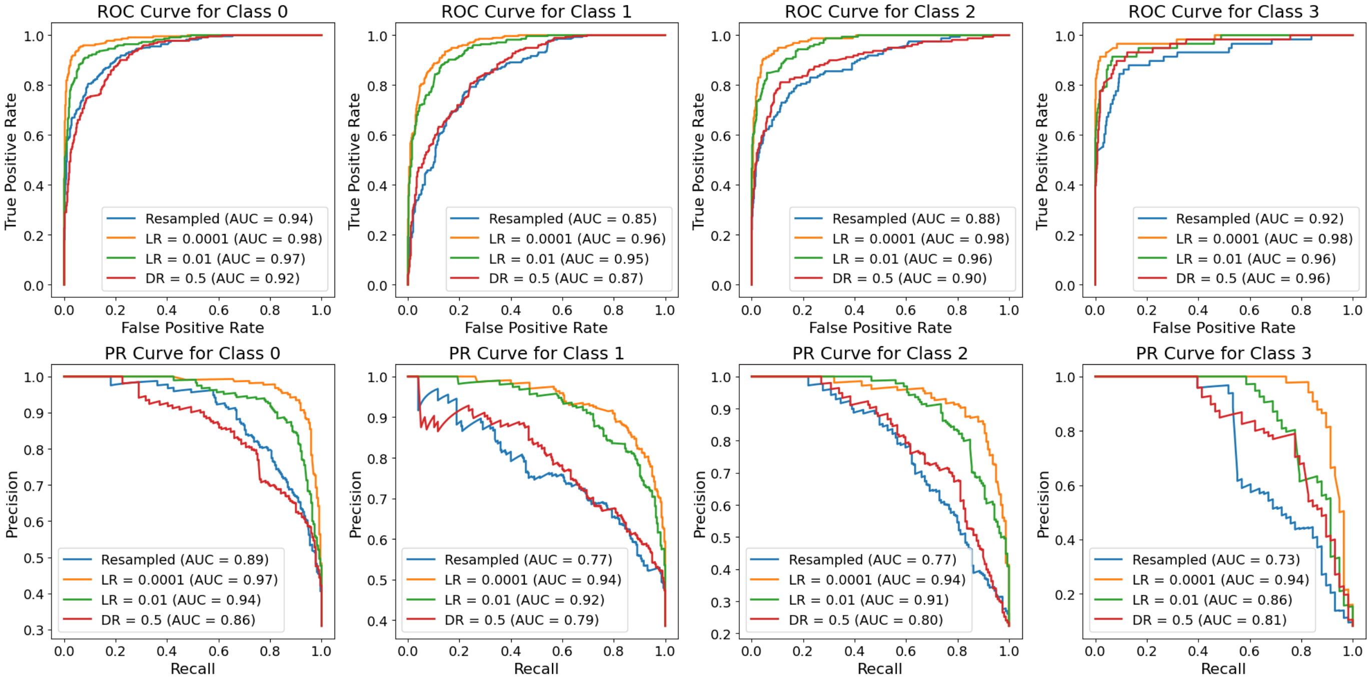


Figure 28: ROC and PR curves for all multi-class femur lateral side WCCE DenseNet-121 models with different hyperparameters. The ROC curve (top) and PR curve (bottom) display the performance of all 4 DenseNet-121 models, per class. Each subplot represents one class, showing how well the models distinguish between the given class and the rest. The class distribution for this subregion can be found in Figure 1.