Osteoarthritis Classification using Convolutional Neural Networks to Mitigate the Digital Divide

Siddarth Vijayan

Abstract

Osteoarthritis, the erosion of cartilage in the knee region, affects millions of individuals around the world, and, if left untreated, can severely hinder quality of life. To support diagnoses of osteoarthritis, convolutional neural networks (CNNs) can supplant doctors' decisions when identifying the severity of osteoarthritis. However, since many CNNs require substantial computing power, this limits their utility to doctors in regions with limited infrastructure, thereby exacerbating the digital divide. In this paper, we implement a custom neural network that uses X-ray images to classify the severity of osteoarthritis that only requires minimal computational resources. We show that through careful hyperparameter tuning, our model can achieve high levels of accuracy even though it has far fewer parameters than many CNNs. Therefore, we anticipate that medical professionals could use our work in resource-limited areas as an aid to diagnosis.

1. Introduction

Over 528 million people worldwide are diagnosed with osteoarthritis, a form of arthritis that affects the protective bone cartilage in joints, more commonly found in hands, knees, hips, and spine. Common symptoms are pain, stiffness, tenderness, and swelling in joints, which can result in an extreme loss of flexibility. Grating sensations can be felt due to bones rubbing together, and bone spurs, or small bone growths, can appear as well [1]. Such severe cartilage degradation causes grinding between the two bones, which results in extreme pain and a lack of mobility. A proper diagnosis is critical and should be made promptly to determine the best course of action for treating arthritis.

While medical professionals are well-trained to make such diagnoses, errors can still occur in hospitals. As a result, there has been a growing interest in applying machine learning to support clinicians in their decision-making, leading to a plethora of models designed to address many issues in hospitals, such as image classification [2]. However, these models are extremely complex and can often be difficult to use in locations or regions with limited computational resources. This "digital divide" has become an increasingly prevalent problem as AI models become more advanced. Areas that lack access to the technical infrastructure required by these more complex models, therefore, have their potential to innovate and grow stymied [3].

Thus, there is a need to develop models of sufficient complexity that are still capable of being run with limited computing power. To address this gap, we develop a parsimonious convolutional neural network that has only 439,589 parameters, compared to other CNNs that can have up to millions of parameters [4]. We use a classical statistical model, logistic classification, to serve as our baseline. While the multinomial logistic classifier has scores of 36% accuracy, the CNN performs significantly better, with scores of 66%, despite using significantly fewer parameters than state-of-the-art CNN models and only requiring a CPU.

Therefore, our model strikes an effective balance between performance and computing resource requirements. This allows it to be used by doctors worldwide, rather than only those in more advanced countries, making these models both effective and accessible to practitioners even in resource-constrained locations.

2. Methods

The inputs of the model are X-rays of the knees of different severities of arthritis. The objective of the model is to classify these images into one of five classes corresponding to varying degrees of arthritis.

2.1. Dataset

Here are some samples from the dataset from class 0, a normal healthy knee, and class 4, severely afflicted by osteoarthritis.

Figure 1

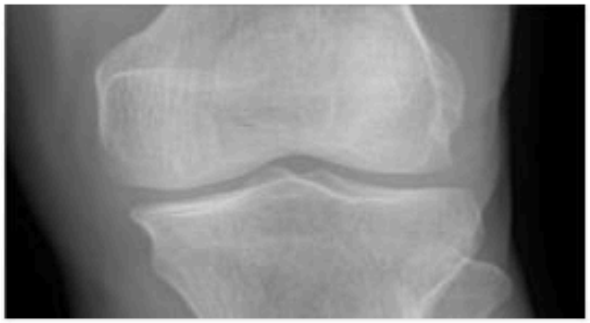

Figure 1A: A healthy knee with cartilage and no joint wear [5]

Figure 1B: Extreme cartilage degradation and bone contact [5]

Figure 1A has a substantial amount of cartilage as indicated by the spacing between the femur (thigh bone) and the tibia (shin bone). However, Figure 1B displays little to no cartilage in the joint, as shown by the close contact between the femur and tibia.

The data used to train the CNN was an image-based dataset with spatial dimensions of 300x162, provided by Kaggle. The dataset contains 1650 images of X-rays of the knee with varying levels of osteoarthritis [5]. However, the number of images in each class was quite unbalanced, so a weighted sampler was used to balance the dataset while collecting images to train the model on each batch.

The dataset was split into the five classes according to the Kellgren-Lawrence grading system, and the table below illustrates the number of samples within each respective class in the dataset.

Table 1: Sample Sizes

Class 0Normal	514
Class 1Doubtful	477
Class 2Mild	232
Class 3Moderate	221
Class 4Severe	206

Table 1: A table containing sample sizes of each class corresponding to each level of severity of osteoarthritis

Various augmentations, such as 15-degree rotations, horizontal image flipping, Gaussian blur applications, and brightness adjustments, were applied to the images as well to make the model resistant to small infinitesimal changes. By applying a weighted sampler for the dataset, the batches being created now had a more equal distribution of classes, instead of one class being favored just because it had more data points.

The dataset was divided using a method called stratified splitting with a ratio of 80-20 between training and testing. As for the validation set to monitor loss convergence, stratified k-fold cross-validation was implemented in order for the model to maximize the total amount of data seen by splitting the existing training set into *k* folds. These folds would then alternate between being the testing validation set and training set, with one fold of data being the testing validation set and the other four folds being the training set.

2.2. Models

2.2.1. Multinomial logistic regression

The baseline model for the X-ray image classification is a multinomial logistic regression model. It is a classical statistical technique that generalizes logistic regression to multi-class problems with more than two simple outcomes, where the log odds of the outcomes are modeled as a linear combination of the predictor variables [6]. The model performed rather poorly, with a total accuracy of 36%.

2.2.2. Convolutional neural network

In this paper, we implement a convolutional neural network [7] (CNN). A CNN is a class of deep learning models that excel particularly in image classification by taking the convolution unit and shifting it across a 2-dimensional array of input values. The convolution process is repeated by adjusting the weights and providing a more in-depth analysis, and returns a higher-level output. By utilizing other techniques like backpropagation, the weights become reinforced, and certain weights will be more resistant to change as training of the model progresses. Information

is passed through the neurons in a forward pass and a backward pass, and each pass adjusts the weights of each neuron.

The model architecture consists of only two simple convolution layers and a fully connected layer that would be replaced with a fine-tuned pre-trained model. Using transfer learning, the final fully connected layer is replaced with a pre-trained model called ResNet50, and its parameters are disabled, or frozen, so that the model does not interfere too much with the existing model architecture.

After each convolutional layer, a batch normalization function is applied to reduce the internal covariate shift, which can lead to slower convergence due to shifts in weights and biases during training in both forward and backward passes. The forward pass through the model first applies the batch normalization, then applies the non-linear ReLU function, and finally pools the layer. Then the output of the last convolution layer is flattened before being put into the fully connected layer, simplifying it into the five classes the dataset has defined. The non-linear function is not applied here because we want the raw output of the model or a real-valued number.

Each layer has a kernel size of 5, meaning the image that the model convolves is a 5 by 5 pixel image, and the final convolution layer expands into 32 channels. The first layer goes from 1 channel, because of the black and white nature of the image, to 16 channels. The output of the first layer is then input into the next layer, going from 16 channels to 32 channels.

Table 2

Layers	Input Channels	Output Channels	Kernel Size	Description
Conv Layer 1	1	16	5x5	Converts grayscale input into 16 features
Conv Layer 2	16	32	5x5	Convert 16 features into 32 features Final conv layer

Table 2: Descriptions of the convolution layer architecture

The model utilizes cross-entropy loss to calculate loss for both the training and validation datasets. The model employs an optimizer algorithm called Adam, which is well-known for its first-order gradient-based optimization of stochastic objective functions. Essentially, the optimizer will adjust the model's weights and parameters in an attempt to decrease the loss or difference between the predicted and true labels of the images. Using the model's parameters, the learning rate was set to 1e-4, and the weight decay was set to 1e-3 to ensure that the model would not overfit. A dropout function of 50% was implemented to randomly shut off neurons in the network as another precaution to prevent overfitting. Here, the ReLU function was applied to the first three convolutional layers and the first fully connected layer; however, the last fully connected layer did not have any ReLU application to preserve the data that represents the scores of each class for the classification.

The model would undergo a training phase and then immediately proceed to a validation set to assess its performance on unfamiliar data. The model then begins to run inside a k-fold

cross-validation loop, with 5 folds, to maximize the use of the limited data in the training dataset. Using stratified k-fold cross-validation, each epoch would contain 5 folds of data that would each serve as a validation set for the model at least once.

Early stopping was implemented to make sure the model would stop training once the model's performance on the validation set begins to degrade, despite any training performance improvements. The function would contain a patience variable and would monitor validation loss levels each fold. Once patience was exceeded, the function stopped the model from continuing to train and overfitting or learning too much from the data. Due to hyperparameter tuning, the model experienced a varying number of epochs; however, the epoch training amount can be multiplied by 5 due to the 5-fold cross-validation method that was used.

2.2.3. Metrics

To assess model performance, we use two metrics: accuracy and F1 score. The accuracy is defined as the number of true positives and false positives, while the F1 score is the harmonic mean of the precision, the number of positive guesses the model made, and recall, the number of positive guesses that were correct, scores.

Results and Discussion

The overall accuracy of the final model was 66% and the accuracy for each class varied. The accuracies, specifically for the first and last classes, were marginally higher than the other classes, and that fact may be due to the model's inability to generalize well (see Table 3 below). The model also vastly outperformed the baseline model despite its limited number of convolution layers.

Table 3: Class Accuracies

Classes	CNN	Logistic Regression
0 - Normal	71.7%	10.9%
1 - Doubtful	68.2%	54.5%
2 - Mild	36.8%	52.6%
3 - Moderate	61.1%	0.0%
4 - Severe	85.7%	66.7%

Out of all the classes, class 2, containing mild images of osteoarthritis, performed the worst, and there are several reasons why this could happen. However, the individual class accuracy does not provide a full in-depth analysis of how many samples the model incorrectly identified or predicted as another class.

A classification report (Table 4 below) is a method for evaluating a model's overall performance on any dataset, whether the model is being trained or tested. Precision is a metric displayed in the classification report of a CNN and quantifies the number of true positives the

model predicted correctly. Its ratio consists of the number of predicted true positives over all predicted positives, including the false positives. The recall metric serves to display the classifier's completeness and consists of a ratio of true positives to the sum of true positives and false negatives. Out of all positive instances, the ratio represents the percentage of predictions that were actually correct. The F1-score is the harmonic mean or average between the precision and recall metrics. The support statistic visualizes how many items the model has seen and can be useful for identifying potential class imbalances; however, in a test dataset, it is not as necessary. Here is the classification report for the initial baseline model.

Table 4: Classification Report on test results - Logistic Regression

	Precision	Recall	F1-Score	Support
0Normal	0.56	0.72	0.75	46
1Doubtful	0.50	0.68	0.68	44
2Mild	0.19	0.37	0.40	19
3Moderate	0.00	0.00	0.00	18
4Severe	0.36	0.67	0.47	21

While the baseline seems to perform decently for classes 0 and 1, classes 2 and 4 perform less favorably, while class 3 received zero correct identification instances during testing. However, the CNN performed much better across the same test set.

Table 5: Classification Report on test results - CNN

	Precision	Recall	F1-Score	Support
0Normal	0.79	0.72	0.75	46
1Doubtful	0.68	0.68	0.68	44
2Mild	0.44	0.37	0.40	19
3Moderate	0.73	0.61	0.67	18
4Severe	0.58	0.86	0.69	21

While the other classes have precision and recall statistics that do not wildly differ from each other, the high recall and low precision of class 4 indicate that the model is prone to classifying X-rays with arthritis as severe even if the diagnosis is only mild, moderate, or even doubtful.

In order to visualize the general performance of the model, a confusion matrix can be used to illustrate the model's tendencies for making predictions. A confusion matrix (Figure 3) displays all of the model predictions along with the true labels for each item in the test set.

Figure 2

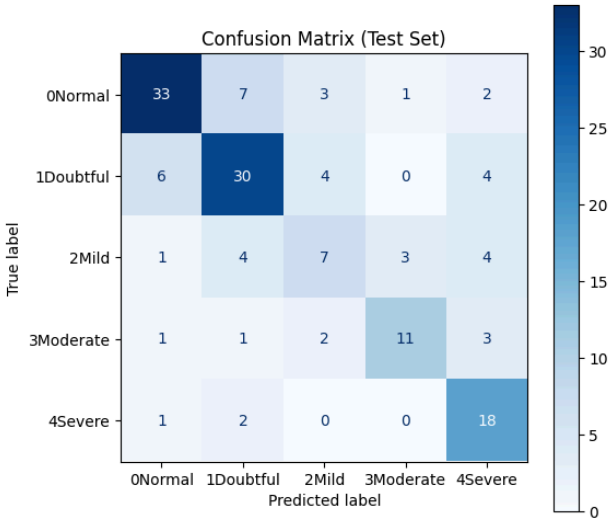

Figure 2: A confusion matrix displaying the overall nature of the model while classifying instances of osteoarthritis.

These scores were achieved through rigorous hyperparameter adjustments and alterations to model architecture; however, the model consistently seems to favor the extreme classes (0Normal and 4Severe). This can be attributed to the fact that osteoarthritis is much more present or apparent in more severe cases, and normal knees have little to no signs of osteoarthritis, which makes the model more confident in its decision. On the other hand, the middle classes perform less well, as the model cannot generalize well enough and reads into smaller details and edges too much, leading the model to make more extreme decisions due to very small differences within each image. As a case study, we consider the images in Figure 3, which display class severities of doubtful and mild, respectively.

Figure 3

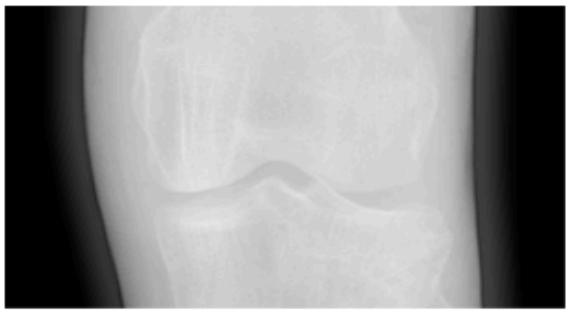

Figure 3A: Instance from class 2, a doubtful case of osteoarthritis

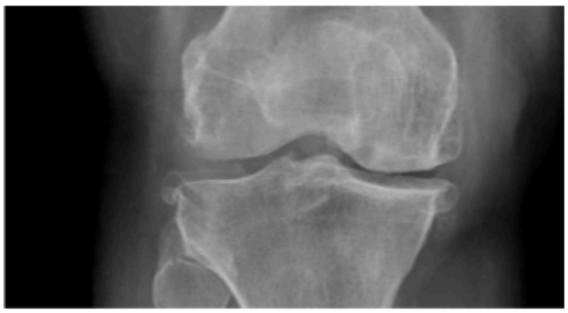

Figure 3B: Instance from class 3, a mild case of osteoarthritis

Due to similarities between the two images, the model may struggle to differentiate between them, which explains the very low performance of class 2, mild osteoarthritis. The

model suffers from being unable to generalize, meaning that the model picks on minute differences “too well”, also known as overfitting, causing the model to make decisions on other parts of the X-rays instead of the actual knee joint area.

Conclusion

In this paper, we demonstrate the implementation of a parsimonious convolutional neural network that is not only computationally efficient relative to more complex CNNs but also exceeds performance on classical statistical baselines. Our CNNs exceed the baseline models in nearly every metric, such as F1-score and accuracy, while maintaining an extremely small amount of trainable parameters.

Future work should focus on hyperparameter tuning and pretrained network tuning to create the ideal model architecture. The model should be further trained in situations where bulk X-rays can be analyzed to determine the existence of osteoarthritis and healthy knee joints, before being applied in situations where the model can exactly identify the class severity of the knee. There is also more to an arthritis diagnosis than just an X-ray, as before the scanning itself, there are meetings between the doctor and patient, discussing how long the pain in the target area has persisted, when it has started, how severe the pain is, etc. By adding more details, the model could definitely improve if given these extra parameters, and it may be the difference between a doubtful case and a severe case. Another area of improvement would be obtaining more data to try to fix the class imbalances and also introduce more X-rays that contain unique features, so the model can learn to generalize better instead of focusing too much on the smaller details. Creating a separate model to analyze the model predictions could also be helpful, as well as a model to consider the patient’s symptoms beforehand, up until the X-ray.

Acknowledgements

I would like to thank Joe Xiao for being my mentor and providing me with advice on how to efficiently construct a convolutional neural network, and checking in on me to make sure I understand the concepts wholly. I would also like to thank Jeremy Bigness for helping me edit and proofread, as well as checking my citations.

References

[1] "Osteoarthritis." n.d. Accessed June 30, 2025. https://www.who.int/news-room/fact-sheets/detail/osteoarthritis.

[2] Rani, Suman, Minakshi Memoria, Ahmad Almogren, et al. 2024. "Deep Learning to Combat Knee Osteoarthritis and Severity Assessment by Using CNN-Based Classification." *BMC Musculoskeletal Disorders* 25 (1): 817. https://doi.org/10.1186/s12891-024-07942-9.

[3] "State of Compute Access: How to Bridge the New Digital Divide." n.d. Accessed July 25, 2025. https://institute.global/insights/tech-and-digitalisation/state-of-compute-access-how-to-bridge-the-new-digital-divide.

[4] Antony, Joseph, Kevin McGuinness, Kieran Moran, and Noel E. O'Connor. 2017. "Automatic Detection of Knee Joints and Quantification of Knee Osteoarthritis Severity Using Convolutional Neural Networks." In *Machine Learning and Data Mining in Pattern Recognition*, edited by Petra Perner, vol. 10358. Lecture Notes in Computer Science. Springer International Publishing. https://doi.org/10.1007/978-3-319-62416-7_27.

[5] Hafiz Nouman. 2024. "🏥Annotated Dataset for Knee Arthritis Detection🦴." Kaggle.com. 2024.https://www.kaggle.com/datasets/7ba24eae2788efeac3948883f42905ee5f218be27fc7f4f383f4657579f24f75/data.

[6] Anderson, Carolyn J, and Leslie Rutkowski. n.d. *MULTINOMIAL LOGISTIC REGRESSION*.

[7] Lecun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haaner Abstract|. 1998. "Gradient-Based Learning Applied to Document Recognition." *PROC. OF the IEEE*. http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf.