# Knee X-Ray Images Classifier

A Multilabel Image Classification Problem

Omi Singh

5/29/24

# Contents

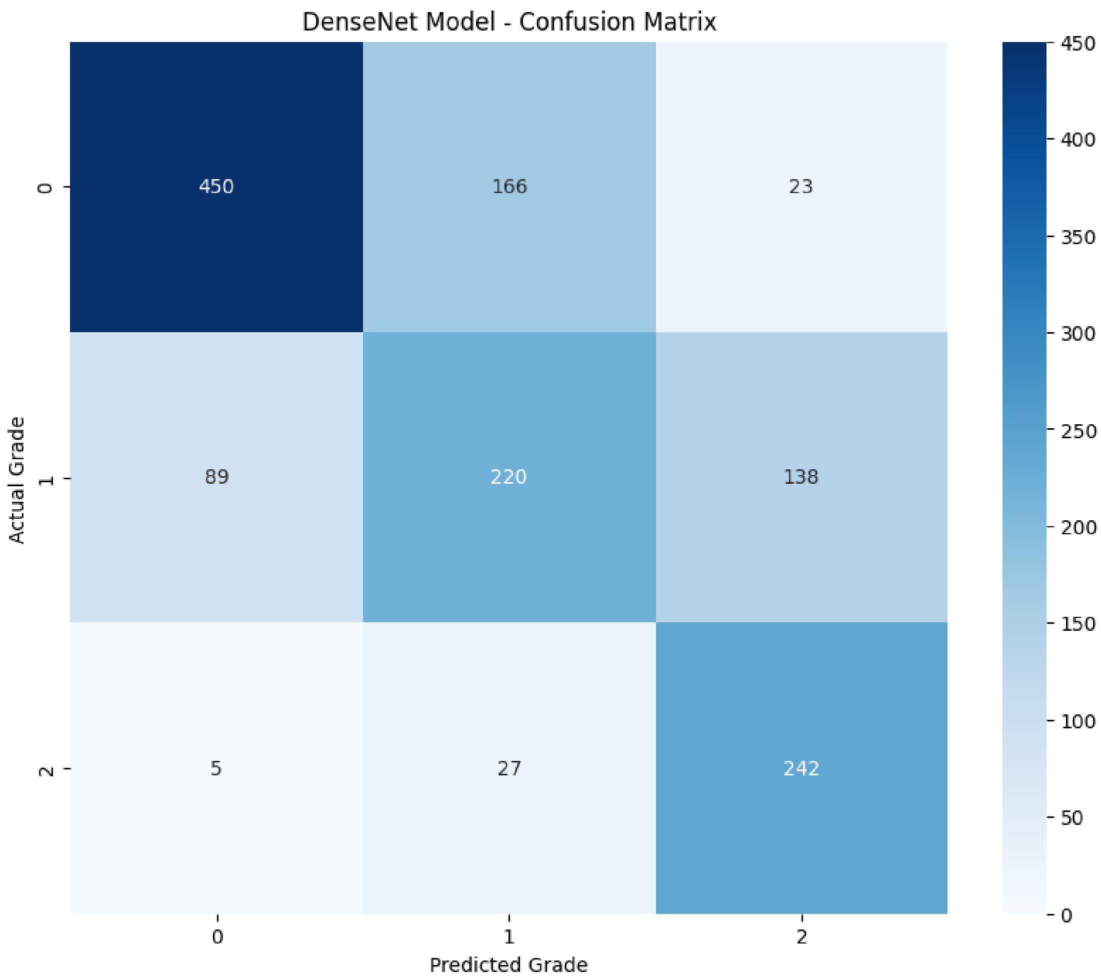
# Summary

The Goal of this Project is to classify Knee X-ray images as Healthy, Low-risk or High-risk (Severe) cases based on osteoarthritis grade.

This was done using 6,765 images.

Two different neural networks were used to build prediction models. A Convolutional Neural Network (CNN) built from scratch and a pretrained DenseNet201 Model.

DenseNet model correctly classified 88% of high-risk test cases.
The model struggled with low-risk cases the most due to their similarity with healthy X-ray images. Model performance can be improved by adding more images or breaking down the problem into binary classification, focused on high-risk cases.




*(Selected model caught 88% of high-risk test cases)*

# The Process

We have followed a standard Data Science Project Methodology. After identifying the goal of the project as a multilabel image classifier.

We have broken down the actual data steps into the four components noted below.

Problem Statement: Using available images build a neural network to classify Knee X-ray images based on arthritis risk.

Data Steps:

- Data Load
- Exploratory Data Analysis (EDA)
- Image Preprocessing
- Model Evaluation and Selection

# Data Load

Our data was taken from Kaggle. The images were transformed into 3 class problems due to the relatively low number of high-risk cases. The original names of grades: Healthy, Doubtful, Minimal, Moderate and Severe also had an overlap in terms of end goal of assessing overall risk for osteoarthritis.

As such the problem was broken down into 3 label problems.

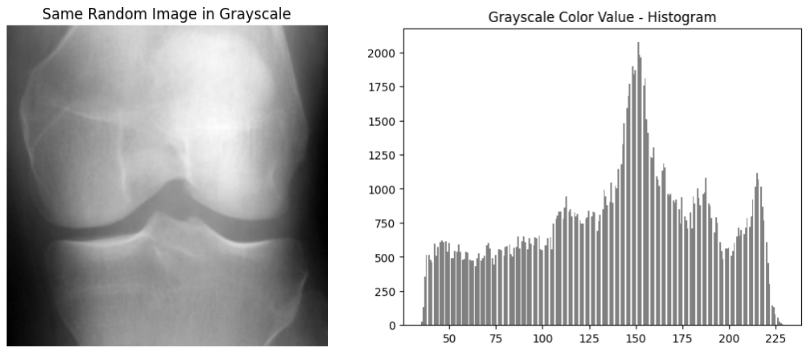
| Knee-Osteoarthritis Severity Grades | |
|---|---|
| **Grade** | **Description** |
| 0 | Healthy |
| 1 | Low Risk |
| 2 | High Risk/ Severe |

During the data loading stage, we also checked for file extensions to make sure images loaded were in .jpeg, .jpg, .png or .bmp formats.

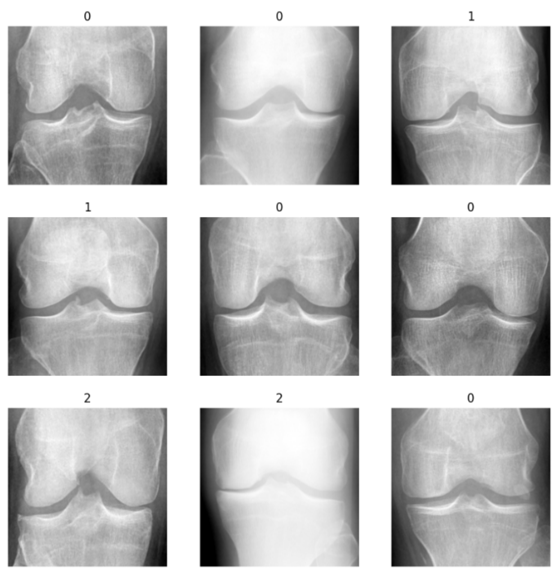
Data Link: https://www.kaggle.com/datasets/shashwatwork/knee-osteoarthritis-dataset-with-severity/data

# Exploratory Data Analysis

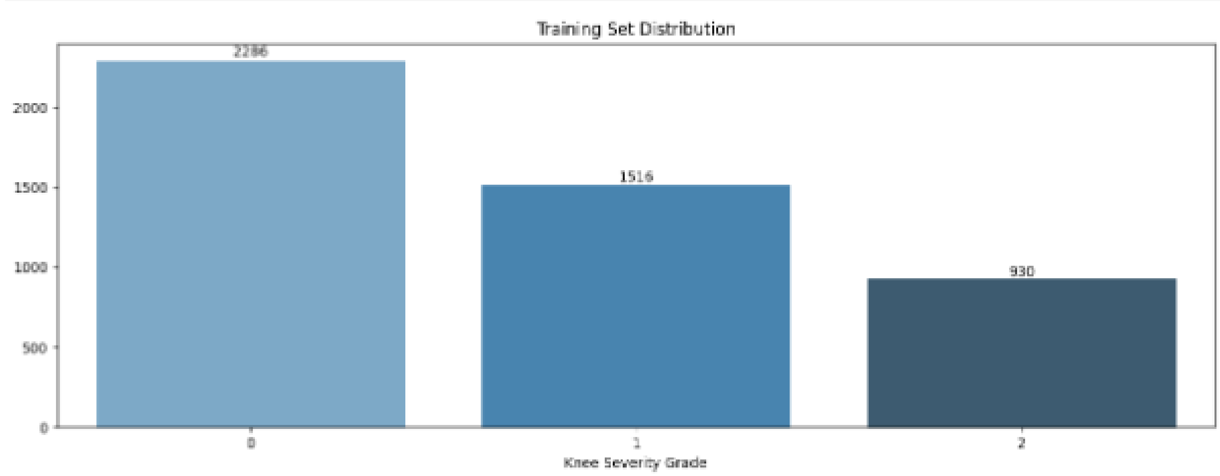
We started by exploring a sample image to make sure the image quality and color values are as expected.




We also explored sample images for all labels. One visual point noted during this stage was that in high-risk cases often there was a reduced gap between the bones




Finally, we looked at distribution of images over the three grades. This led us to understand discrepancy in data size for three labels. This discrepancy was addressed during preprocessing stage using class weight dictionary.



# Image Preprocessing

During the image preprocessing stage, we created data generators to rescale images. The images were loaded in batch size of 32 and image size of 224 x 224 using image data generator.

We also calculated class weights to be later used during model training. A higher-class weight indicates lower overall image count and tells model to account for it by using its inbuilt oversampling technique.

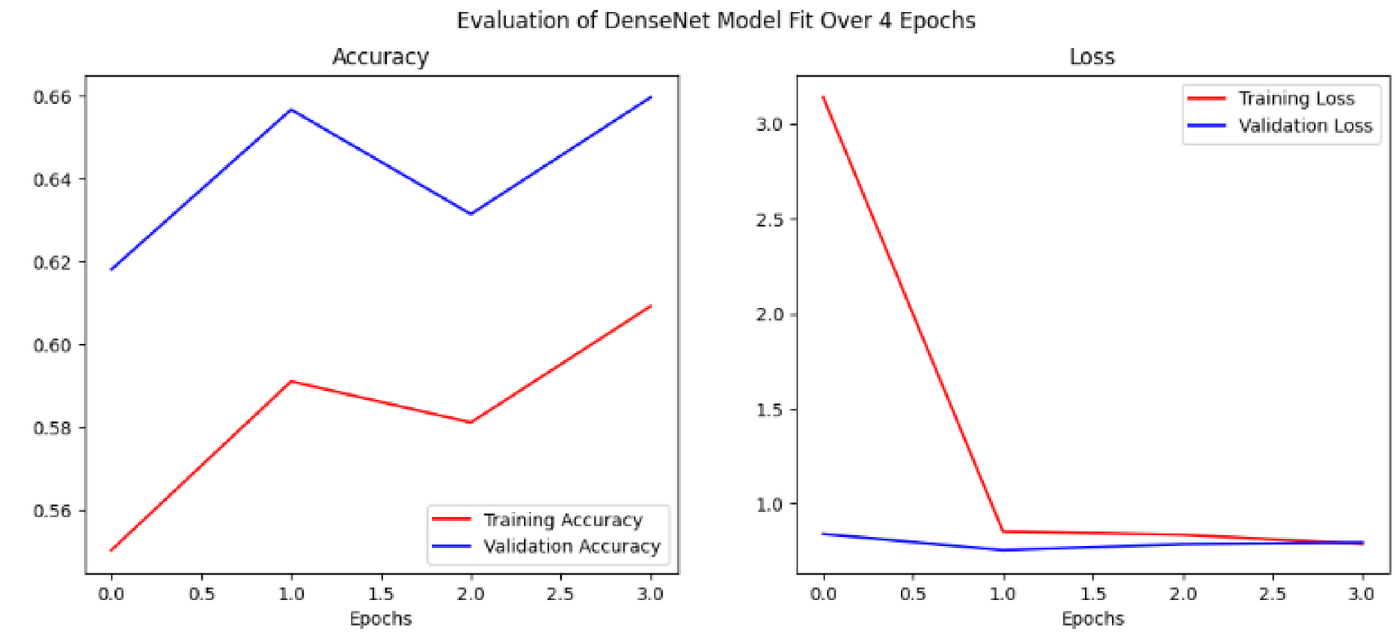
| Class | Class Weight |
|---|---|
| **Healthy** | 0.69 |
| **Low Risk** | 1.04 |
| **High Risk/ Severe** | 1.7 |

This was done for the train, validation and test sets separately, each with their own image generators.

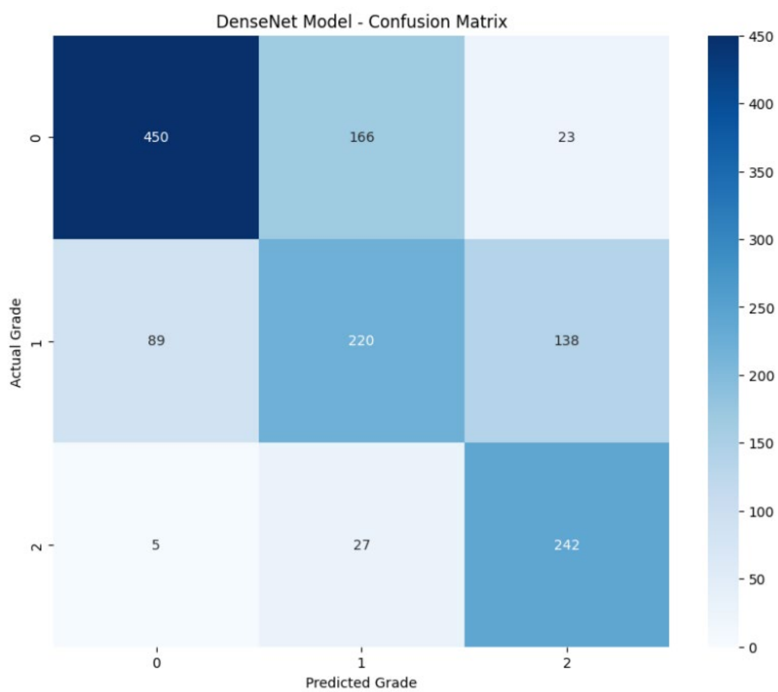
# Modeling – Evaluation and Selection

During modeling stage training and validation generators were used to evaluate a user Convoluted Neural Network (CNN) model and a pre-trained DenseNet201 model.

While the CNN model gave the best validation accuracy of 49% before being stopped by early stopping callback after two subsequent epoch runs didn't show an improvement. The DenseNet model reached the best validation accuracy of 66%.




As such our final model selected was the DenseNet Model.



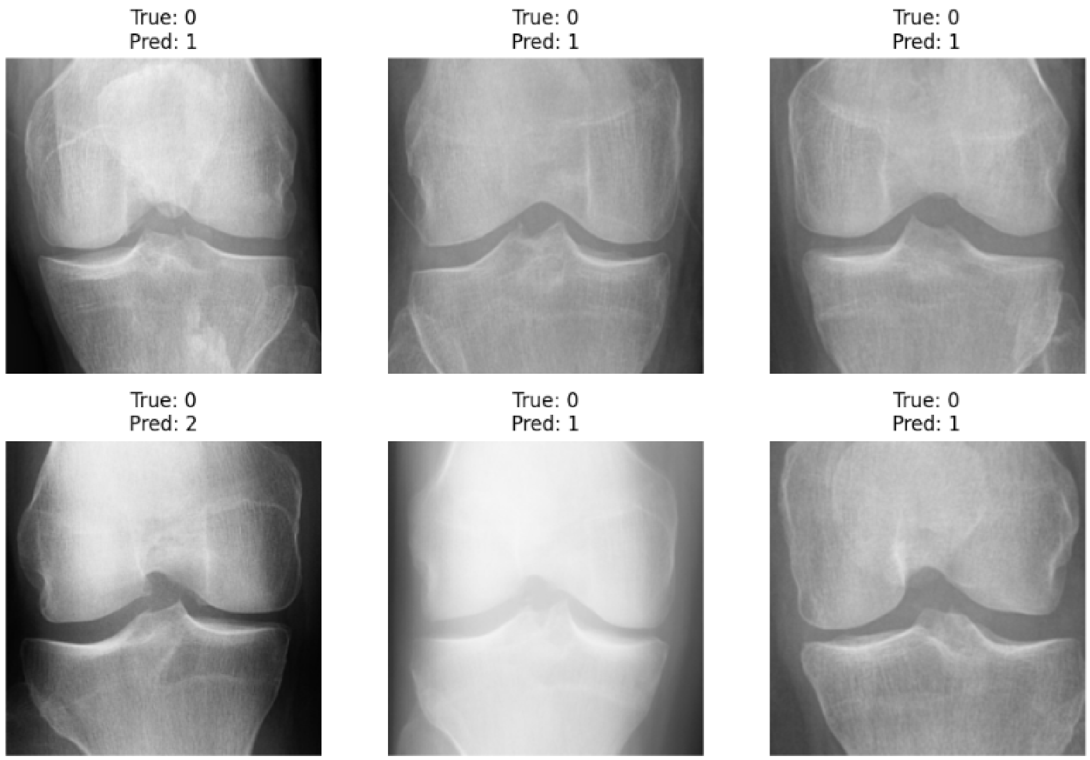

Dense Model - Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.83 | 0.70 | 0.76 | 639 |
| 1 | 0.53 | 0.49 | 0.51 | 447 |
| 2 | 0.60 | $\boxed{0.88}$ | 0.71 | 274 |
| accuracy | | | 0.67 | 1360 |
| macro avg | 0.65 | 0.69 | 0.66 | 1360 |
| weighted avg | 0.68 | 0.67 | 0.67 | 1360 |

# Conclusion

Our image dataset was hard to classify as all images were Knee X-ray images with few distinctive features for each class, especially the low-risk images were hard for the model to decipher.

These low-risk images could be grouped with healthy images to zone in on identifying high risk cases. Since we used 4,732 training images adding more images will also help model improve performance for low-risk cases.




Given the constraints it is quite impressive that the pretrained DenseNet201 model was able to correctly catch 90% high risk cases.

Keras API includes several pretrained image neural network classifiers with their individual strengths and applications noted. Deep Learning has been made quite accessible for industries, thanks to advancements in these APIs.