Detecting Abnormalities in Musculoskeletal X-Rays: Project Milestone

Adisa Kruayatidee
Department of Computer Science
Stanford University
adisa@stanford.edu

Abstract

To meet the needs for an automated means of detecting musculoskeletal conditions, this project explores binary classification of bone x-rays as normal or abnormal. It differs from previous attempts by experimenting with a single models that are relatively cheap and explainable. The hybrid ResNet-ViT model outperforms similar experiments using a CNN-only architecture. While its performance doesn't measure up to previous studies, it has shown potential for improvement.

1. Introduction

Musculoskeletal conditions affect over 1.7 billion people worldwide. They are the leading cause of severe, long-term pain and disability, with 30 million emergency room visits annually and increasing. These conditions are made up of a variety of diagnoses, ranging from simple fractures to longer-term conditions like osteoarthritis.

Conditions are often diagnosed by trained radiologists examining bone x-rays. However, the limited number of trained radiologists, especially in developing countries, makes it increasingly critical to have an automated, efficient means of detecting bone abnormalities from x-rays. This project takes as input a bone x-ray from one of seven upper-extremity categories (elbow, finger, forearm, hand, humerus, shoulder, wrist). It passes it through a hybrid Convolutional Neural Network (CNN) and Vision Transformer (ViT) model, and outputs a prediction of whether the bone is normal or abnormal.

CNNs have been used extensively for medical classification tasks with promising results, sometimes exceeding specialist performance. Their architecture makes them naturally efficient at feature extraction. More modern transformers have also gained immense popularity in recent years. They are able to capture global context and longer-range dependencies across an image. Combining the two should yield a more expressive model.

2. Related Work

The Stanford ML group did the foundational research on the MURA (**mu**sculoskeletal **ra**diographs) dataset that they produced [1]. Their architecture was a 169-layer Densely Connected Convolutional Network (DenseNet), followed by a fully connected layer with one output, and a sigmoid nonlinearity. The DenseNet was initialized with weights from a model pretrained on ImageNet, and models were trained end-to-end using an Adam optimizer to minimize the weighted binary cross-entropy loss. The top five models with the lowest validation losses were ensembled for the final model. See Table 1 for comparisons of their Cohen's Kappa score, compared to certified radiologists.

Many other teams have attempted to improve upon the Stanford baseline. A large number of approaches also used different flavors of CNN-based architecture. However, like [2], many failed to match performance of the baseline, further highlighting the difficulty of abnormality detection. Eclipsing the baseline often required computationally expensive ensemble models like [3], [4]. Any single model (deep CNN) that matched performance is still expensive [5]. Coming up with such a model also required extensive experimenting that isn't necessarily theoretically motivated, with various optimizers, feature extraction methods, and architectures [6].

More recently, teams have expanded beyond primarily CNN models. [7] was the inspiration for attempting a hybrid CNN-ViT architecture. I wanted to see if I could achieve similar results using a Residual Network (ResNet), as they are more efficient than DenseNets [8]. ResNets have proven successful in other binary classification tasks, such as detection of Covid-19 in CT images [9].

Besides experimentation with training different model architectures, there have also been interesting studies with different learning techniques, such as test-time augmentation [10] and self-supervised learning [11].

Study Type	Worst Radiologist	Best Radiologist	Stanford Baseline
Elbow	0.710	0.850	0.710
Finger	0.304	0.410	0.389
Forearm	0.802	0.796	0.737
Hand	0.661	0.927	0.851
Humerus	0.733	0.933	0.600
Shoulder	0.791	0.864	0.729
Wrist	0.791	0.931	0.931

Table 1. Comparisons of Cohen's kappa score.

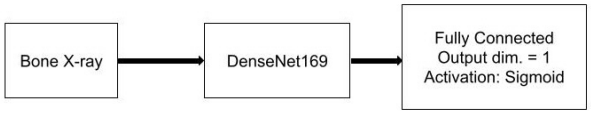

Figure 1. Densenet with simple classifier

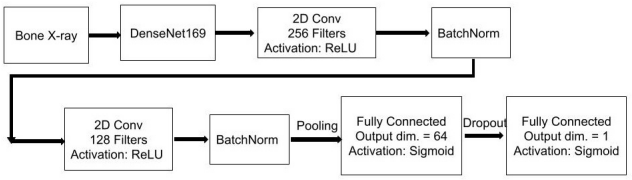

Figure 2. Densenet with convolutional classifier

3. Methods

3.1. Transfer Learning

Transfer learning is a technique wherein a model developed for a task is reused as the starting point for a model on a similar, second task. Instead of training the CNNs used in my models from scratch, starting with random weights, I initialize them with weights from a model that has been trained on a large, general dataset (ImageNet). Although the ImageNet images are very different from bone x-rays, theoretically, the pre-trained model has learned to extract fundamental, general features like edges, textures, and shapes. Besides saving computation and training time, transfer learning is important in this application because MURA is a relatively small dataset, and a CNN trained from scratch may not be able to extract features as well.

3.2. Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs) are a specialized class of deep neural networks primarily designed to process data with a known, grid-like topology (images!). They have revolutionized fields like computer vision due to their ability to automatically learn hierarchical features directly from raw pixel data. See Figures 1 and 2 for my CNN architectures with various classifier heads.

3.3. Hybrid CNN-ViT

Vision Transformers (ViTs) adapt the highly successful Transformer architecture to image analysis. Unlike traditional CNNs which rely on convolutional layers, ViTs leverage self-attention mechanisms to understand visual information.

As demonstrated in the work by Hussain et al. [7], the use of hybrid models can improve accuracy and sensitivity/specificity in comparison to CNN-only models. Theoretically, the CNN does feature extraction, and the ViT is able to record global dependencies over the entire image. I implemented a similar model using the TensorFlow Keras library.

3.4. Loss Functions

I primarily used Binary Cross-Entropy Loss. For a batch of N samples, the total loss is defined as

$$-\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)]$$

where y_i is the true label (0 or 1) and $\hat{y}_i$ is the predicted probability for the i-th sample in the batch.

Given the imbalance between positive and negative examples in the training set, I also tried Facebook AI's Focal loss. It is a modification on standard cross-entropy loss:

$$-(1-p_t)^{\gamma}\log(p_t)$$

γ is a tunable parameter, greater than or equal to zero. p_t is the predicted probability of the true class. If p_t is high, meaning the example is well-classified and "easy", then $1-p_t$ will be small, and raising it to a power will make it smaller. This down-weights the contribution of easy examples to the loss. Similarly, if p_t is low for a "hard" example, raising it to a power will not change its contribution to the loss much.

3.5. Optimizers

The Adam optimizer defines an adaptive learning rate per parameter. It tries to move faster in directions with con-

sistent gradients and slower in directions with noisy or inconsistent gradients:

$$
\begin{aligned}
m_t &= \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \\
v_t &= \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \\
\hat{m}_t &= \frac{m_t}{1 - \beta_1^t} \\
\hat{v}_t &= \frac{v_t}{1 - \beta_2^t} \\
\theta_t &= \theta_{t-1} - \frac{\alpha \cdot \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}
\end{aligned}
$$

In contrast, Stochastic Gradient Descent with momentum takes steps in the direction of the mini-batch gradient, but it also accumulates a "velocity" (momentum) from previous gradients. This velocity helps accelerate the gradient descent in the relevant direction:

$$
\begin{aligned}
v_{t+1} &= \gamma v_t + \eta \nabla_\theta J(\theta) \\
\theta_{t+1} &= \theta_t - v_{t+1}
\end{aligned}
$$

4. Dataset

MURA (**mu**sculoskeletal **ra**diographs) is a publicly available dataset produced by the Stanford ML group [1]. It contains about 40,000 multi-view bone x-rays from more than 14,000 studies conducted at the Stanford Hospital between 2001 and 2012. The studies are grouped into seven categories as mentioned above. Each study was hand-labeled as normal or abnormal by a board-certified radiologist while viewing on a high-resolution, medical-grade display. The original x-rays were acquired with a native resolution of 1500x2000 pixels, but the clinical images in the dataset vary in resolution and aspect ratio.

The dataset was already split into training and validation sets. The training set had 21935 normal and 14837 abnormal images. The validation set was (presumably intentionally) more balanced, with 1665 normal and 1532 abnormal images. Since the Stanford ML group did not disclose their test set, I randomly reserved 30% of each category of the validation set to use as a test set.

4.1. Data Preprocessing

Since the first layers of each model I experimented with used networks pre-trained on ImageNet, the x-rays had to be preprocessed to be similar to ImageNet [12]. I scaled each image to 320x320. The square aspect was not too different from the average/median aspect ratio of the x-rays (0.8). I also normalized each image to have the same mean and standard deviation as ImageNet images.

5. Experiments, Results and Discussion

The architectures with which I experimented the most were the pretrained DenseNet169 ("baseline" model) and the pretrained ResNet50 + ViT ("hybrid" model).

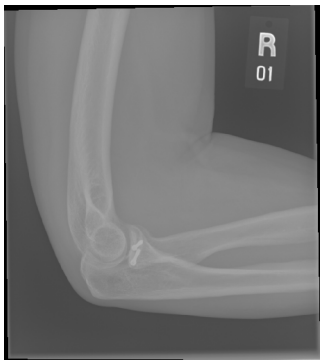

Figure 3. Abnormal Elbow

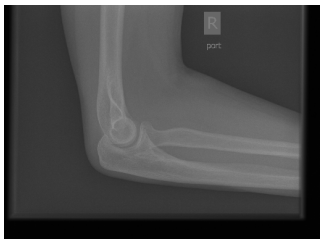

Figure 4. Normal Elbow

5.1. Training Baseline Model

First, I tried training several variants of the baseline, using a simple fully-connected layer with sigmoid nonlinearity as the classifier head. I initially thought the problem was overfitting, since the model could get pretty high (above 0.90) training accuracy even after little training, but the validation accuracy was stuck. I applied regularization techniques to combat this: adding random augmentations to the training batches (horizontal flip, moderate amounts of rotation, zoom, contrast) so it would be harder for the model to "memorize" data; freezing the DenseNet params to reduce the model's potential expressivity; including pooling and dropout layers in the classifier head; adding L2 regularizer to the classifier weights. These served to bring the training accuracy down, but the validation accuracy was curiously still the same.

I then inspected the models' predictions on the test set, and saw that they never predicted an abnormal X-ray. The real problem was that the models were suffering from majority class bias, which made sense, given that there were about 1.5 times as many normal as abnormal X-rays in the training data. The models were resorting to an "easy" prediction, instead of learning about the data. To encourage abnormal predictions and more strongly penalize false normal predictions, I tried two different techniques, adding class weights and using a focal loss function. To help the model generalize to unseen data, I tried using a smaller batch size (8 instead of 32), since the "noise" introduced by smaller

batch gradients might help the model escape narrow local minima. Since the problem could also have been with the classifier's lack of expressiveness, I replaced the fully-connected classifier with a convolutional classifier.

None of these changes, applied alone or in combination, helped the baseline model to predict abnormal X-rays. As a sanity check, I also trained a model from scratch, without using the ImageNet weights, and it similarly got stuck predicting only normal X-rays. The breakthrough was in changing the optimizer to use SGD, instead of Adam. I had not thought to use an optimizer besides Adam, since all related works I read reported using Adam with no issues. If the gradients coming from the minority abnormal class were initially small, noisy, or sparse (e.g. abnormal features are subtle), Adam's adaptive nature might have caused it to reduce the effective learning rate for parameters related to detecting abnormalities. Predicting all normal X-rays was a quick path to reducing the overall loss, in comparison to the difficult task of distinguishing abnormalities. Unlike Adam, SGD applies a consistent step size to all parameters, which helps it "push across" the loss landscape and escape local minima. Combined with the class weights, SGD could translate the abnormal class' amplified loss signal into a consistent push across the entire network.

5.2. Training Hybrid Model

After the baseline model started showing more reasonable performance in predicting abnormalities, it began overfitting to the training set. I had trouble effectively regularizing it (increasing L2 and dropout, and applying more aggressive mixup data augmentation). Overfitting can be caused by too many model parameters, which motivated using a smaller CNN in my hybrid model. I chose a ResNet50.

I first froze the ResNet and trained only my classifier head, with an initial learning rate of 0.00019 and class weights boost factor 1.5. I obtained these hyperparameters by running an Optuna study. Optuna is an open-source, automatic hyperparameter searching framework. It runs more efficiently than random search by estimating a promising area of the hyperparameter space in each trial, and pruning unpromising areas.

Once the performance started stagnating, I unfroze the ResNet. I lowered the learning rate to 0.000001 to avoid drastically changing the ResNet weights on each gradient update, thus "forgetting" what it had learned. From experience in the baseline training, I used SGD as the optimizer. Since my learning rate was essentially a guess, I applied two callbacks monitoring the validation AUC in the training, to avoid fruitlessly training a model which is not improving after a certain number of epochs: early stopping and learning rate reduction.

It seemed that the model was mostly stagnating, so I started increasing the learning rate, which yielded improvement. However, after more epochs, I noticed an interesting phenomenon, that the training accuracy was starting to decrease slightly, even as the validation accuracy and AUC were increasing. This indicated training stability, so I decided to switch optimizers to Adam. I also noticed that both training and validation recall were low, so I increased the class weights boost.

After the model started plateauing in both training and validation accuracy, I ran another Optuna study to see if a better combination of learning rate and class weights boost could yield better results. The various Optuna trials had pretty similar results, indicating that the model had reached its capacity to learn.

5.3. Quantitative Results

5.3.1 Metrics

Using this confusion matrix for binary classification:

	Predicted	
Actual	Positive	Negative
Positive	TP	FN
Negative	FP	TN

I'll evaluate models using the following metrics:

- Cohen's kappa statistic = $\frac{p_0 - p_e}{1 - p_e}$, where

  $p_0 = \frac{\text{TP+ TN}}{\text{N}}$ (observed agreement)

  $p_e = \frac{\text{TP + FP}}{\text{N}} \cdot \frac{\text{TP + FN}}{\text{N}} + \frac{\text{TN + FN}}{\text{N}} \cdot \frac{\text{TN + FP}}{\text{N}}$ (agreement by chance)

  N = TP + TN + FP + FN (total number of samples)

- Sensitivity = $\frac{\text{TP}}{\text{TP + FN}}$
- Specificity = $\frac{\text{TN}}{\text{TN + FP}}$

I will also calculate the Area Under the Curve (AUC) metric, which measures the performance of a binary classifier by quantifying its ability to distinguish between positive and negative instances across various probability thresholds. It is a rough measure of accuracy, which is more indicative than the actual accuracy metric for imbalanced datasets. For example, a model that always guesses "negative" would achieve 95% accuracy on a dataset that has 95% negative samples, but it is not necessarily a good model because it cannot distinguish positive samples). An AUC of 1 represents perfect classification, while 0.5 is no better than random guessing.

5.3.2 Results

See Table 2. While Cohen's Kappa did not come close to the Stanford group, it is encouraging that the hybrid model

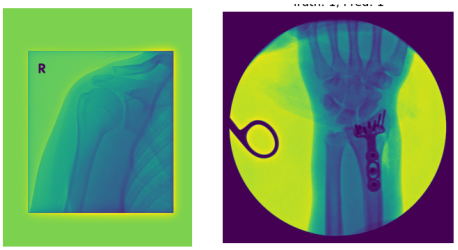

Figure 5. True vs. false positive, color contrast

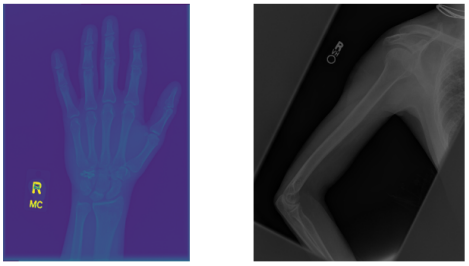

Figure 6. True vs. false negative, correct alignment

outperformed my baseline CNN-only model in most categories, attesting to the power of transformers. Potentially, it did less well in the elbow studies because long-range dependencies that aren't related to an abnormality, like the angle of the elbow, confused it. For its lower performance in the finger studies, similarly, long-range dependencies across the image might not be as relevant, if finger abnormalities are quite small. The fact that the AUCs are relatively high also suggests room for easy improvement, like choosing different prediction thresholds for each category.

5.4. Qualitative Results

See figures 5 and 6 for comparisons of correctly classified (left image) vs. misclassified (right image). One reason for false positives is that color-contrast within the bone could be a strong signal for an abnormality, but the color contrast may also arise from variance in the x-ray machinery. False negatives are harder to judge, because the abnormality might be subtle, especially if larger patterns like skeletal alignment look normal.

6. Conclusion and Future Work

If I had more time, my goal would be to try to surpass the Stanford group's metrics. Backing up, I could also try to implement and train the Stanford group's original baseline, to try to get close. This would potentially involve a larger hyperparameter sweep. I need better strategies for debugging models when their performance isn't improving, as well as more compute power to train a larger hybrid model (bigger ResNet, more transformer layers). It would also be interesting to experiment with additional strategies that were mentioned in the Related works section, like self-supervised learning and test-time data augmentation.

7. Contributions

This was a one-person project. All aspects of this project were handled by the author. This project did not use any public code nor involve non-CS231N contributors.

References

[1] Pranav Rajpurkar, Jeremy Irvin, Aarti Bagul, Daisy Ding, Tony Duan, Hershel Mehta, Brandon Yang, Kaylie Zhu, Dillon Laird, Robyn L. Ball, et al. MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs. *arXiv preprint arXiv:1712.06957*, 2017.

[2] Dennis Banga and Peter Waiganjo. Abnormality detection in musculoskeletal radiographs with convolutional neural networks (ensembles) and performance optimization. *arXiv preprint arXiv:1908.02170*,2019.

[3] Goodarz Mehr. Automating Abnormality Detection in Musculoskeletal Radiographs through Deep Learning. *https://arxiv.org/pdf/2010.12030*,2020

[4] Kwun Ho Ngan, Artur d'Avila Garcez, Karen M Knapp, Andy Appelboam, and Constantino Carlos ReyesAldasoro. Making densenet interpretable a case study in clinical radiology. *medRxiv, page 19013730*, 2019.

[5] Li, Xiaoke and Li, Yang and Qi, Lin and Wang, Yanjie Enhanced deep residual network for bone classification and abnormality detection *Computational Intelligence and Neuroscience*, 2022

[6] Prakash U M, Arivazhagan N Deep Convolutional Neural Network Based Detection of Bone Abnormalities in Musculoskeletal Radiographs. *International Journal of Intelligent Systems and Applications in Engineering*, 2024

[7] Hussain, Muhammad and Khan, Muhammad Moazam and Tariq, Usman and Armghan, Muhammad and Khan, Muhammad Adnan and Rehman, Asif and Khan, Muhammad Awais and Kadry, Seifedine. A Hybrid Convolutional and Vision Transformer Model with Attention Mechanism for Enhanced Bone Fracture Detection in X-Ray Imaging. *Sensors*, Vol. 23, No. 15, p. 6686, 2023.

[8] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Memory-efficient implementation of densenets. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5584–5592, 2017.

Study	Baseline AUC	Baseline Cohen's Kappa	Hybrid AUC	Hybrid Cohen's Kappa
Elbow	0.784	0.353	0.773	0.533
Finger	0.651	0.222	0.659	0.144
Forearm	0.743	0.382	0.763	0.381
Hand	0.620	0.129	0.702	0.292
Humerus	0.595	0.260	0.720	0.336
Shoulder	0.575	0.002	0.736	0.269
Wrist	0.710	0.291	0.778	0.363

Table 2. Baseline and Hybrid metrics comparison

[9] Ali Abbasian Ardakani, Alireza Rajabzadeh Kanafi, U Rajendra Acharya, Nazanin Khadem, and Afshin Mohammadi. Application of deep learning technique to manage covid-19 in routine clinical practice using ct images: Results of 10 convolutional neural networks. *Computers in Biology and Medicine, page 103795*, 2020.

[10] Amiri, Mohammad Sadegh and Hajimirsadeghi, Hamed and Mashayekhi, Hamed and Taheri, Seyed Mostafa. Improving convolutional neural networks performance for image classification using test time augmentation: a case study using MURA dataset. *BMC Medical Imaging*, Vol. 24, No. 1, p. 61, 2024.

[11] Behzad Bozorgtabar, Dwarikanath Mahapatra, Guillaume Vray, Jean-Philippe Thiran. Anomaly Detection on X-Rays Using Self-Supervised Aggregation Learning *arXiv:2010.09856*, 2020

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee*, 2009.