RiskRay: Detecting hips at risk of fracture using machine learning

Motivation

Millions of people suffer from hip fractures worldwide every year. They are a hallmark fracture of osteoporosis, with 70-90% of fracture cases stemming from this disease. According to a 1993 study, the first year of costs for a patient with a hip fracture in the US is $26,000 USD[1].

The current gold standard for detecting if a hip is at risk of fracture is using Dual Energy X-ray Absorptiometry, “DEXA”. The procedure involves using two different energies of x-rays, and without describing the entire process, produces a measurement of bone density. This measurement, however, is known to have significant variation (as much as 30%) across its many manufacturers and instantiations. Access to these expensive machines is limited, and may have patients waiting weeks or months for their appointment, which cost around $106 CAD per test, in 2006.

With all of this in consideration, and the fact that this technique was developed and deployed in the 1960’s, we believe there is an opportunity for this technology to be superseded. Our machine learning architecture presents an approach that is not only more accurate, but would be significantly faster and cheaper, with technicians/doctors easily executing the utility in tandem with a patient’s hip x-ray.

Architecture

Neural Network Model

Supervised machine learning (ML) is a style of ML where data is labeled by a human (or other means) from a limited set of labels. The ML model is then tuned or “trained” to identify the label(s) on new data previously unseen. Ultimately, it is a form of pattern recognition, and part of our investigation is to answer the question: *Is there* *visual information contained in x-ray images of at-risk hips vs. healthy hips that could be picked up by a ML model*? When optimized, ML image recognition can notice the subtlest of signatures and could be able to pick up on details that evade even a professional's eye. We aim to also incorporate an x-rays metadata into the network, since information like subject's age will aid in the model's assessment.

Getting a labeled dataset is often the road block in a supervised learning pipeline. We introduce a novel key insight for building up this dataset. If a hip fracture occurs on one side of the body, then the opposite hip can be marked as an at-risk hip (conditional on hip health being at least in some part responsible for the fracture). These at-risk hips that are the counterparts of fractured hips can form the at-risk portion of the dataset. We then complete the dataset with images of x-rays of healthy hips.

With this method, we compiled a dataset of 113 at-risk hip x-rays and 360 healthy, "control", hip x-rays, for 473 images total which were divided as 378 Training images, and 95 Test images (20 at-risk, 75 healthy/control). Indeed, this approaches the lower bound of an acceptable dataset size in this context, but is enough to be worth the experiment.

We used a Mixed Data neural network (MDNN) architecture that had two main components, followed by a concatenating layer. The first main block was a regular image processing network of two Convolutional Neural Networks (CNNs) and a Multilayer Perceptron (MLP) layer. The second was a small MLP applied to image metadata including the patient age and a small number of x-ray fields: kVP, distance-to-detector, and exposure time. The output from these two main channels were concatenated in a layer that passed through another MLP before outputting the final logistic result in the range [0, 1] (at-risk or control). By default, values >0.5 are approximated as at-risk, and those <=0.5 are considered control. In future iterations of this architecture, adding more fields to the second block of metadata fields would be interesting. For example, the answer to "Is there a history of osteoporosis in your family?" Finally, a 3rd party open source software package Ray Tune was used to systematically optimize hyperparameters across various runs.

Final RiskRay Product

The final RiskRay product will consist of 3 main components that function to intake a hip x-ray/DICOM, and output a normalized category of risk, a "Risk Rating" or "Risk Score", for the user. The components are:

1. An X-ray/DICOM cropping software utility that technicians can use to click & drag the hip region in a patient's x-ray. This cropped portion is input into the MDNN for the next phase.
2. The MDNN intakes the hip image and outputs a its classification in the range [0, 1], where 0 is towards healthy hips and 1 towards at-risk hips.
3. The classification is further categorized using the mapping in Table 1 to output a normalized Risk Rating.

It's worth noting that the network's output in the range [0, 1] is quite natural, really, as even humans often only have a partial degree of certainty in their estimations. Note that the Risk Rating categorization matrix in Table 1 is a suggestion and is subject to change.

Output	Risk Rating
[0, 0.4]	Healthy
[0.4, 0.6]	Moderate
[0.6, 1.0]	At risk

Table 1. MDNN Output to Risk Rating

Results & Discussion

Performance

Over the course of searching the hyperparameter space with RayTune, over 300 trials had been run, evaluating small (~5e3 parameters) to larger network sizes (~5e6 parameters). Of course, 5e6 parameters is still modest, but performance seemed to plateau before this size anyway. In this section we will look at a single run from the larger network which performed well, though its results are not unique to this run.

The model evaluation was performed on 95 (20 at-risk) images unseen during training. I will spare explaining all the evaluation metrics in full detail, but will offer brief descriptions. Models were evaluated against a thorough suite of metrics. Throughout, a “Positive” result is the model outputting an estimate that is greater than some threshold (which is by default 0.5), and in our case indicates an At-risk hip:

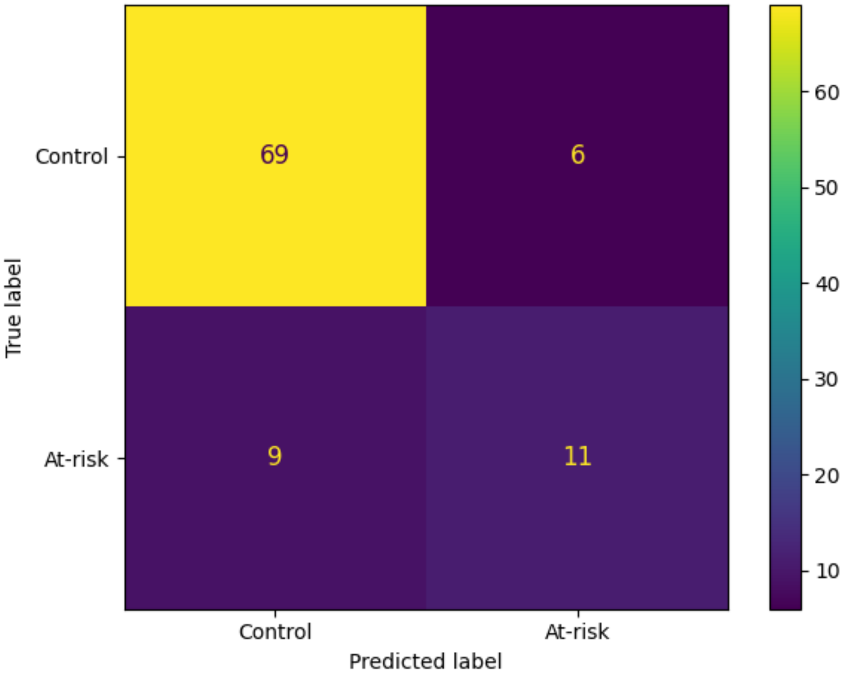
1. Precision: *What fraction of our estimated Positives were True Positives?*
2. Recall: *What fraction of our At-risk hips were detected as Positive?*
3. F1: Harmonic mean of Precision and Recall together. Range is [0,1].
4. Confusion Matrices: A snapshot of the distribution of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN).
5. ROC & AUC: The probability that a positive sample will be ranked higher than a negative sample.

In our case, we are inclined to value Recall over Precision. These two metrics are at odds in practical contexts. High precision indicates a very selective, stringent model that tries to minimize False Positives (healthy hips reading as at-risk). However, this comes at the cost of letting at-risk hips slip through unnoticed (increased False Negatives, low recall). High recall indicates that we are doing well at capturing at-risk hips, but this looser restriction lets more False Positives through. In medical applications, high recall is often preferred so positives are not missed, with the downside of potentially treating false positives. It is not a zero-sum game however, and better models will indeed score better at both precision and recall than a lesser model (which is neatly captured in the F1 score).

ROC and AUC are common metrics for classification estimators like ours. The ROC curve, as will be shown, plots True Positive Rate (TPR) vs False Positive Rate (FPR) *as the classification/decision threshold is swept from [0, 1]*. By default, the threshold is 0.5—an estimate of 0.6 gets rounded to 1 as Positive result and vice-versa. However, an estimator may *still be* an effective discriminator, separating positives from negatives, but about a different threshold. For example, it estimates positives to be roughly in the range [0.3, 0.4] and negatives to be [0.1, 0.2]. At a threshold of 0.5, this model performs terribly, but not at a threshold of 0.25. If an estimator perfectly separates its test dataset, the ROC curve will be a square, TPR=1 constant (See our figure below for a sense of this). AUC (Area under the curve) would be its maximum at 1. In reality, there is mixing of FPs and FNs, which reduces the curve away from a square, AUC<1. The worse this mixing, the worse your AUC, and a slope of 1 is as good as random estimation.

First, we'll look at the sample run's TP, FP, FN, TN distribution (Fig 1), and the Precision, Recall, F1-score (Table 2). By Fig 1, the model shows a bias towards the Control class, with only 6 of 75 Controls being FP. It would be better to see more of the At-risk images correctly identified, 9 of the 20 were FN, lowering our recall. The Positive precision was somewhat better with only 6 of 20 being FP. Overall, this this model correctly labeled 80 of 95 images, giving an accuracy of 84%, and a "balanced" accuracy (the mean accuracy by class) of 74%.

Fig 1: Confusion matrix



The Precision, Recall, and F1-score are given in Table 2. “Support” is the number of images of that type in the Test dataset. “Macro Average” is the flat average of the metric values, and “Weighted Average” weights the metrics by their Support count.

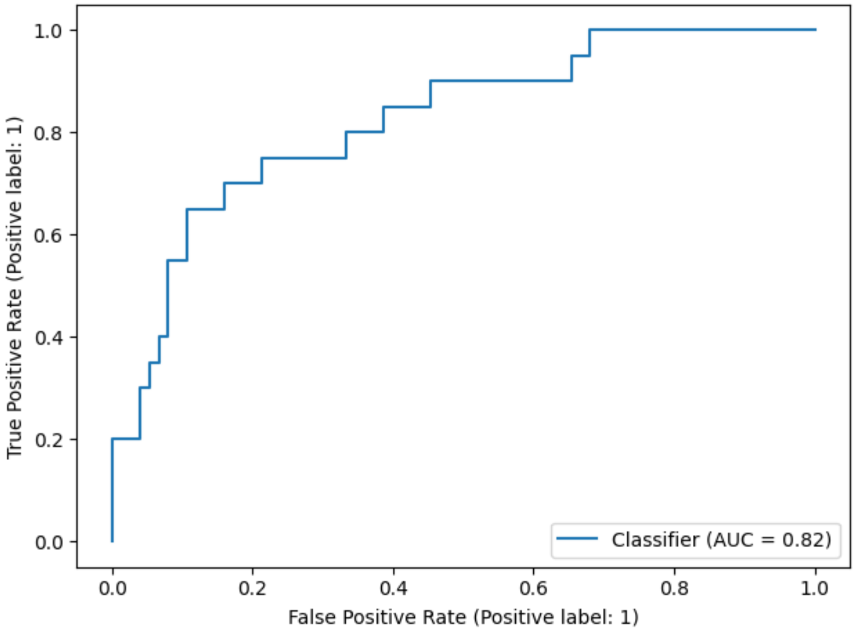
Table 2. Performance metrics on selected model

Label	Precision	Recall	f1-score	Support
Control	0.88	0.92	0.90	75
At-risk	0.65	0.55	0.59	20
Macro Avg	0.77	0.74	0.75	95
Weighted Avg	0.83	0.84	0.84	95

Figure 2 displays the ROC curve for our sample. There is not so much insight to be gathered form the form itself, beyond that it directs towards TPR=1 as it should, but an AUC of 0.82 is a good sign.

Overall, from these results it stands to reason that this network is indeed learning some feature present in the dataset. Again, the purpose of this study was to see if there was “something there” worth pursuing, and to that end it is successful.

Fig 2. ROC



Sample Inspection

It is also worth getting a visual on what the model evaluates on given images. In the images below, Figures 3 & 4, two batches of images are taken from the Test set for a visual inspection on network output. For the given input images, the model output percentages are overlain in red text along with the category of image: “Ctrl” or “Risk”. Most of the output samples are ideal, approaching the “Healthy” Risk Score for the control batch, and the “At Risk” Risk Score for the at-risk batch. However, exceptions can be found in either batch. Part of this is likely from the limited number of images to train our model. However, it is important to remember the inherent, and limiting, chaos of the data—at least in this pilot context. From the data gathered for this study, there is contamination across both control and at-risk groups because, for example, a hip might be *in reality* at risk, but still be in the control set because the patient hasn’t fractured their hip yet. Conversely, a control hip could be in the at-risk group if the patient’s fracture was due to external circumstances and not hip-health per se.

References

[1] Sernbo and Johnell., 1993. Osteoporosis International 3, 148-53

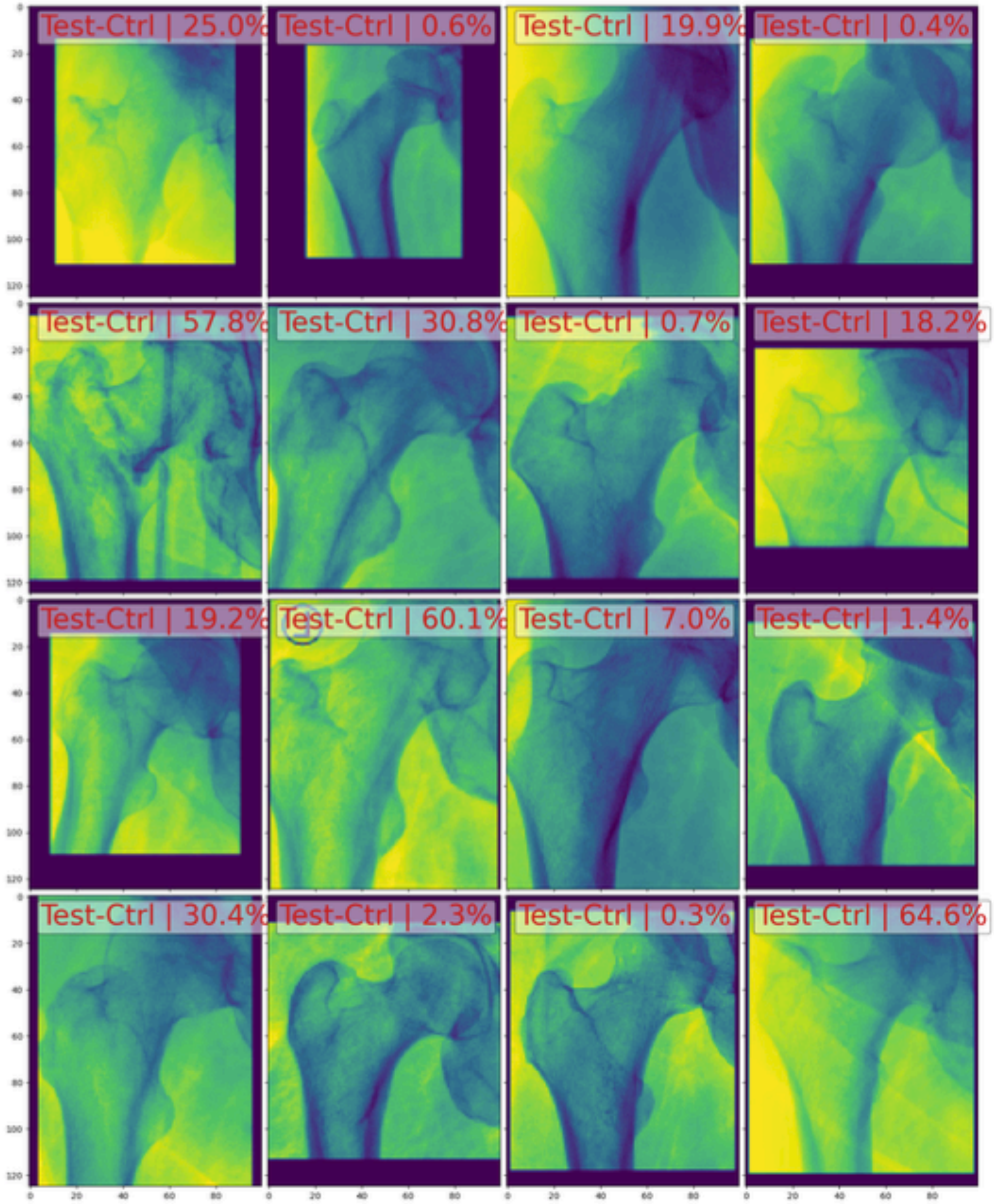


Fig. 3. Test-set control images and the selected model estimations.

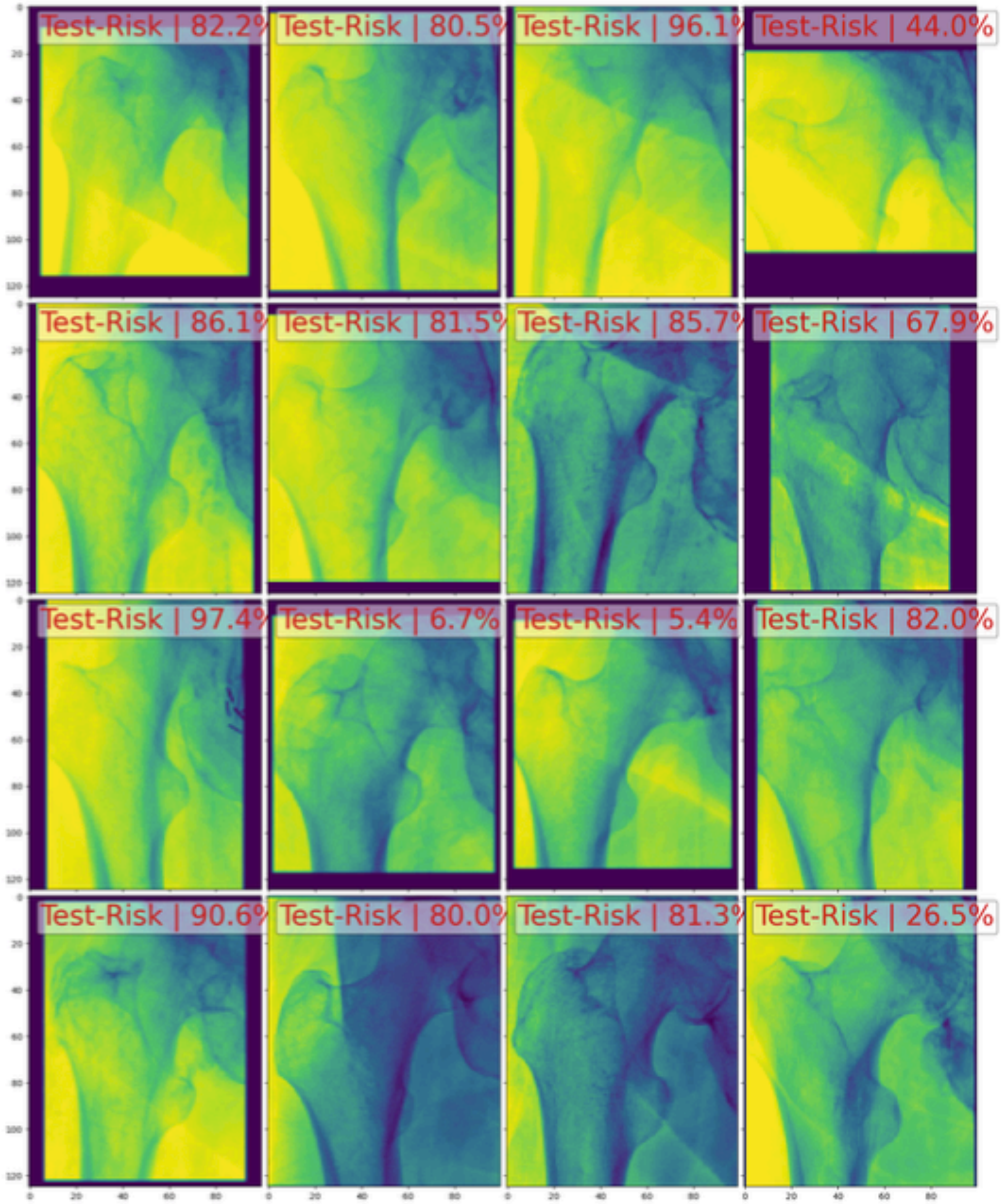


Fig 3. Test-set at-risk images and the selected model estimations.