FractuVision: Enhancing Bone Fracture Diagnostics

Jun Wang
Stanford University
jun2026@stanford.edu

Aryan Siddiqui
Stanford University
aryansid@stanford.edu

Abstract

In our project, we propose solutions for automating the classification and detection of bone fractures in X-ray images. Unlike previous work which heavily relies on supervised learning, we leverage self-supervised learning (SSL) to enable our models to more effectively capture the most relevant features in the images. We use the FracAtlas dataset, which comprises high-quality, annotated X-ray images of bone fractures, to ensure our model's generality and robustness. For fractured/non-fractured classification, our approach begins with a ResNet feature extractor coupled with a Multi-Layer Perceptron (MLP), achieving an initial accuracy of 66.5%. We significantly advance this performance by fine-tuning the DINOv2 self-supervised vision transformer models, which boost classification accuracy up to 96.1%. For bone fracture localization tasks, we finetune the YOLOv8 object detection model, achieving a mean Average Precision (mAP50) of 0.53, demonstrating precise fracture localization capabilities. Our experimentation with SSL-powered bone fracture classification and detection demonstrates the potential of deep learning techniques to significantly enhance clinical diagnostic practices by automating the analysis of X-ray images.

1. Introduction

1.1. Motivations

In recent years, deep learning models have demonstrated major breakthroughs in the field of computer vision. For instance, convolutional neural network (CNN) architectures, and more recently Vision Transformers (ViT), have achieved near perfect performance on classifying common images [14] [4]. We realize that these deep learning models can potentially be applied to the field of medical imaging diagnosis. Currently, many such imaging diagnosis services, such as radiology, have very high equipment and labor costs as doctors typically spend a long time analyzing the images obtained. The deployment of an accurate deep learning model for X-ray image analysis may significantly reduce the time and human labor involved in the diagnostics stage and may drastically reduce the costs of radiology services, making accurate diagnostics much more affordable and efficient to the public.

1.2. Problem Statement

In this project, we focus on the problem of bone fracture detection in musculoskeletal X-ray images in particular. We aim to build a deep learning model which, given a musculoskeletal X-ray image, is able to:

- *Classification*: Make a judgment as to whether the X-ray image contains a bone fracture.
- *Detection*: If the image does contain a fracture, compute bounding box coordinates that correctly identify the location(s) of the fracture(s) within the image.

1.3. Overview of Major Results

For the classification task, we have (1) used self-supervised learning (SSL) to train a Residual Network (ResNet) feature extractor and then trained a Multi-Layer Perceptron (MLP) to make class predictions based on the extracted features, and (2) finetuned the DINO Self-Supervised Vision-Transformer Model to the task of bone fracture classification in X-ray images. The former model achieves a classification accuracy of 66.5%, while the latter achieves a classification accuracy of up to 96.1%.

For the bone fracture detection (localization task), we have finetuned the YOLOv8 object detection model to predict bounding box coordinates for bone fracture spots in an X-ray image. The finetuned model achieves a localization precision of 77%, recall of 55%, and an mAP50 of 0.53. Fractures occuring at thinner bone sections and near body joints present the greatest challenge to the detector.

2. Related Work

The past decade has seen a lot of interest in applying deep learning to X-ray imaging analysis. Deep learning models have been trained to classify chest X-rays [2], perform segmentation of major vessels in X-ray coronary angiography, [16], perform forensic X-ray age estimation [9],

and detect concealed items in cargo radiographs [7]. The application of deep learning in bone fracture classification and detection is a relatively recent endeavor. In 2020, Yadav et. al trained a deep CNN model to distinguish X-ray images of healthy bones from those of fractured bones, and their model achieved a classification accuracy of around 92%. [15] The first known attempt to build a bone fracture detector model was in 2021, when Nguyen et. al finetuned a pretrained YOLO model to detect arm bone fractures in particular, and their model achieved an AP (Average Precision, see 4) of 0.819. [11] In 2022, Hardalacc et. al trained a few regions with convolutional neural network (R-CNN) models with slightly different architectures to detect fractures in wrist bones in particular, and their Faster R-CNN model achieved an AP50 of around 0.58. [6] In the same year, Guan et. al built a novel two-stage region-based CNN model for detecting thigh bone fractures in particular, achieving an Average Precision of 88.9%. [5] Also in 2022, Kong et. al trained a CNN-based model they named DeepSurv to predict spine bone fractures, which achieved a C-index (area under RoC curve) of 0.612. [8] Most recently, in 2023, Sahin built a model that uses edge and corner detection to extract features from X-ray images and then experimented with multiple different classifiers to classify the images as fractured/non-fractured, achieving an accuracy rate of 88.67%. [13]

We notice that the vast majority of current work on bone fracture image classification have followed a supervised approach. Sahin's recent attempt inspires us to explore integrating a feature extractor into a classification model, as the feature extractor can be trained on a large dataset and can also help the model focus on the most relevant information in an X-ray image. Furthermore, to the best of our knowledge, none of the work on automated bone fracture classification and detection so far leverages a transformer-based model, which also encouraged us to make such an attempt.

3. Data

Existing X-ray datasets are either small or lack proper annotation, which hinders development of robust machine learning algorithms for X-ray images. To address these limitations, we use the FracAtlas dataset [1]. FracAtlas contains 4083 musculoskeletal X-ray images collected from three major hospitals in Bangladesh. 720 of these X-ray images contain 922 instances of bone fractures, whereas the rest do not contain any bone fracture instance. The labels were created by two expert radiologists and an orthopedist. Each image is given labels indicating the presence of fracture(s), the count and type(s) of fracture(s) present, as well as bounding box coordinates indicating the location(s) of the fracture(s) in the X-ray image. FracAtlas also provides a division of the 720 X-ray images with fractures into a training set of 575 images, a validation set of 62 images, and a test set of 83 images. We similarly divide the 3363 X-ray images not containing fractures into a training set of 3000 images, a validation set of 163 images, and a test set of 200 images. Hence our dataset is organized as follows:

- Fractured (720 images)
  - Training (575 images)
  - Validation (62 images)
  - Testing (83 images)
- Non-Fractured (3363 images)
  - Training (3000 images)
  - Validation (163 images)
  - Testing (200 images)

3.1. Preprocessing

For the bone fracture detection task, we decide to finetune a pretrained Single-Stage Object Detector named YOLOv8 [10]. In the original FracAtlas dataset, the bounding box locations are provided in a Coco JSON file. In order to make these label data compatible with the YOLO model finetuning process, we used the Pylabel package to convert the Coco JSON file into YOLO text annotation files.

For the image classification task, our baseline model involves a Residual Neural Network (ResNet) feature extractor trained using self-supervised learning and a Multi-Layer Perceptron (MLP) that computes the class scores from the extracted features. The feature extractor is trained using the Lightly Self-Supervised Learning Framework (`https://docs.lightly.ai/`). We realized that the X-ray images in the FracAtlas dataset have only one (grayscale) color channel and very high resolution, which is required for accurate diagnosis of bone fractures. However, Lightly was designed to work with everyday images and thus typically supports standard RGB images of much lower resolutions. Therefore, prior to pretext task training with the Lightly framework, we had to use ImageMagick to convert the X-ray images into the 16-bit TIFF format and resize them so that their length and width dimensions do not exceed 512. As we will discuss later, we suspect that this preprocessing step may have led to the loss of important detail information in the images and compromised the performance of the feature extractor model.

4. Methods

4.1. Binary Classification

4.1.1 Baseline: Self-Supervised Learning Using ResNet and MLP

Our first objective is to classify an X-ray image as to whether it contains a bone fracture. As a baseline, we

trained a model of our own design to perform this binary classification task. When considering suitable model architectures, we realized that a successful model needs to have general knowledge about the fundamentals of X-ray images in order to extract relevant information to inform its classification. Therefore, we decide to leverage self-supervised learning (SSL) to train a feature extractor that is able to compute effective feature embedding vectors from X-ray images. The embedding vectors are then fed into a Multi-Layer Perceptron (MLP) in order to compute the class scores.

One of the few open-source, general purpose SSL frameworks that is widely used is Lightly. This computer vision framework helps in understanding and filtering raw image data and can be applied before any data annotation step, producing image embedding vectors that can be applied to all different types of downstream tasks. Considering our computing resources, we decided to use the full fractured training set and non-fractured training set to train Lightly's default ResNet Feature Extractor, an architecture previously shown to perform well on image classification tasks [14]. The image augmentation operations provided by the Lightly framework include random cropping, horizontal/vertical flip, rotation, gaussian blur, color jittering, random grayscale, and solarization. Due to the grayscale nature of X-ray images, we believe that color-oriented augmentation operations are not relevant for our specific task and hence did not apply them in our SSL process. Furthermore, we also realize that most bone fractures occur on a local level without large-scale traces on the skeleton, making the image details essential for judging the presence of fractures. Therefore, we chose not to apply gaussian blur when training the feature extractor in order to preserve information about details in the X-ray images in the feature embeddings.

Due to the limited size of our dataset, we believed that the MLP model could easily be susceptible to overfitting. Therefore, we decided to include several regularization measures to mitigate the overfitting issue. First, after each fully connected (Linear and ReLU) layer, we decided to add a batch normalization layer to control the scales of the output of the fully connected layer. Second, we included an L2-penalty term for the weights of the fully connected layers to control the scales of the weights. Third, we applied dropout after each fully connected layer to avoid the co-adaptation of parameters and encourage the model to learn more general patterns. The MLP architecture therefore consists of a sequential arrangement of Linear-ReLU-Batchnorm-Dropout blocks. We decided to make the number of such blocks, as well as the L2-regularization parameter and the dropout rate our hyperparameters which we can tune based on the validation set results.

Finally, we noticed that in the FracAtlas dataset, there are far more X-ray images without fractures than there are images containing fractures. Therefore, training the MLP model on the full training set may encourage the model to predict "non-fractured" much more often than "fractured" since non-fractured images significantly outnumber fractured images in the training set. However, a reliable model for potential deployment needs to be able to make inferences purely based on the information contained in the image itself. To address this issue, while we still train the ResNet feature extractor on the full training set, when training the MLP classifier we decided to take only a portion of the non-fractured training images so that they do not significantly outnumber the fractured images. This way, we avoid tempting the model into blindly predicting the more common class.

4.1.2 Supervised Learning using Vision Transformer

Building upon the insights gained from our initial self-supervised approach, we explored the potential of the Vision Transformer (ViT) model for the classification of bone fractures in X-ray images. We wanted to explore the impact of self-attention mechanisms on the feature discrimination capabilities within bone X-ray images. This initial investigation employing supervised learning aimed to set a reference point for subsequent, more complex self-supervised learning experiments involving attention mechanisms.

The Vision Transformer [4] adapts the transformer architecture—originally designed for natural language processing tasks—to the realm of image classification. Unlike traditional CNNs that process images through localized convolutional filters, the Vision Transformer treats an image as a sequence of patches. Each patch is embedded and then processed in a manner similar to tokens in NLP, enabling the model to apply self-attention across these patches. This mechanism allows ViT to dynamically focus on the most informative parts of an image.

For our implementation, we utilized the 'vit-base-patch16-224' model from Hugging Face. We fine-tuned this pre-trained model on our dataset of fractured and non-fractured X-ray images. During the process, we partitioned the images into patches of 16x16 pixels each, with the input images resized to 224x224 pixels, which were then linearly embedded. Positional encodings were added to retain spatial information. The transformer's encoder, consisting of alternating layers of multi-head self-attention and multilayer perceptrons (MLP) blocks, employs layer normalization before each block and residual connections around each to enhance training stability and performance.

4.1.3 Self-Supervised Learning Using DINOv2

To enhance the precision of our fractured/non-fractured classification capabilities in bone X-ray images, we inte-

grated DINOv2 [12], a leading-edge self-supervised learning framework developed by Meta AI. DINOv2 has demonstrated superior performance compared to multimodal text-image models such as CLIP, particularly in tasks focused on visual feature extraction. Motivated by this advanced technology, we sought to evaluate how DINOv2's sophisticated mechanisms would perform in comparison to our earlier custom architecture and the supervised implementation of the Vision Transformer.

DINOv2 advances the concept of knowledge distillation within a self-supervised learning framework using a student-teacher architecture, where both networks are based on the Vision Transformer model. The framework is structured around two primary learning objectives: at the image level, the student network is trained to replicate the teacher network's output from different crops of the same image, promoting a robust understanding of global image features. At the patch level, the student model is trained on randomly masked patches of the image, improving its capability to interpret local details.

In our implementation of DINOv2, we first chose the Vision Transformer Small (ViT-S) as the foundational architecture. We will refer to this as DINOv2 small. We chose ViT-S to balance between computational efficiency and effective feature extraction capabilities. We also explored the impact of using a larger model by incoporating the Vision Transformer Large (ViT-L). We will refer to this as DINOv2 large. Our hypothesis was that a larger model could potentially capture more complex patterns and provide higher accuracy in fracture classification, despite its greater demand on computational resources. Among the three available pretrained heads—image classification, depth estimation, and semantic segmentation—we naturally chose the image classification head.

Typically, there is a two step training process: initial self-supervised training followed by task-specific fine tuning. The initial phase involves both the student and teacher networks learning to extract features autonomously from unlabeled data. However, as mentioned above, we utilize a pre-trained model where this initial self-supervised learning has already been completed. For the fine-tuning phase, we leverage the image classification head that is specifically designed for the task of categorizing images. We directly apply this image classification head to fine-tune the pretrained model on our labeled dataset of bone X-ray images.

4.2. Bone Fracture Detection

Our second objective is to build a model that, given an X-ray image known to contain bone fractures, provide bounding box coordinates that indicate the location(s) of the fracture(s) present in the image. We first considered using Regions with Convolutional Neural Networks (R-CNN), an architecture that uses a convolutional neural networ (CNN) to extract a feature map and a region proposal network to identify regions likely to contain objects. [3] However, we realized that building our own R-CNN model from scratch may involve training the both the backbone convolutional neural network and the region proposal network on a very large image dataset, which is not feasible given our project duration and resources. Furthermore, we do not currently know of a particularly successful open-source pretrained R-CNN model. Therefore, we decided to consider other architectures with pretrained models suitable for finetuning to the specific task of bone fracture detection in X-ray images.

One of the most popular pretrained models for object detection is YOLO, a Single-Stage Object Detector released in 2016 which was trained on a large image dataset featuring a wide range of different objects. The latest model released, YOLOv8, achieves state-of-the-art object detection performance for common images. Although object detection in everyday images is very different from bone fracture detection in X-ray images, there is currently no detection model pretrained specifically for the purpose of bone fracture detection to our best knowledge, largely due to the difficulty to obtain large musculoskeletal X-ray image datasets. Furthermore, through the pretraining process, we believed that the YOLO model would be able to learn the characteristics of bone fractures effectively based on its prior knowledge on general object detection. Thus, we decided to use the fractured training set to finetune the YOLOv8 model for bone fracture detection.

To evaluate the performance of our finetuned model in identifying the locations of fractures, we chose to use the mean Average Precision metric. This metric is based on the Intersection over Union (IoU) of each predicted bounding box, which is defined as the ratio between the intersection and union of the predicted bounding box and the ground truth bounding box.

$$IoU_{\text{predicted box}} = \frac{\text{Overlap with ground truth}}{\text{Union with ground truth}}$$

The mAP50, for instance, gives the fraction of predicted bounding boxes with an IoU of at least 50%.

We also create a confusion matrix to evaluate the performance of our finetuned detection model. As is consistent with the threshold conventions of YOLO models, we define the successful identification of a fracture as an overlap with an IoU of at least 0.45 between a ground truth bounding box and a predicted bounding box with a confidence level of at least 0.25. [1] Therefore, a ground truth bounding box that does not have such an overlap with a predicted bounding box is considered a false negative, and a predicted bounding box that does not have such an overlap with a ground

[1] https://docs.ultralytics.com/reference/utils/metrics/#ultralytics.utils.metrics.ConfusionMatrix

truth bounding box is considered a false positive.

5. Experiments

5.1. Classification: Self-Supervised Learning using ResNet and Multi-Layer Perceptron

As a baseline model for the binary classification task, we used the Lightly SSL framework to train a ResNet feature extractor with pretext tasks and then trained a Multi-Layer Perceptron (MLP) to compute the class scores from the extracted features. For the pretext task training, we used the fractured training set (575 images) and the non-fractured training set (3000 images) and included random cropping, rotation, and horizontal/vertical flips as image augmentation operations. Lightly applies these augmentations randomly to the images for pretext task training. To make the image format consistent with the Lightly framework, we converted the images used for self-supervised learning into 16-bit TIFF format and resized them so that their length and width dimensions do not exceed 512. We trained the feature extractor for 100 epochs with the pretext tasks and saved the best model. Then, we used the best model to generate feature embeddings for all images in the FracAtlas dataset.

The embedding for each image is a vector of dimension 32. The MLP was trained to compute the class score of an image given its embedding vector. To train the MLP, we used the first 1000 of the 3000 non-fractured training images and the 575 fractured training images. This ensures that in the training set for the MLP, the non-fractured images do not overwhelmingly outnumber the fractured images, thus avoiding tempting the model to blindly predict the more common class. The fractured and non-fractured validation sets are used for validation. Originally, we experimented with a model containing five Connected-Batchnorm-Dropout blocks with a dropout rate of 0.5 and an L2 regularization parameter of 0.001. However, we found that the model was significantly overfitting to the training set as the validation loss began to increase very early in the training process. Therefore, we reduced the number of blocks and increased the dropout rate and regularization parameter. We were able to obtain the best validation performance with a model containing two blocks, with a dropout rate of 0.65 and an L2 regularization parameter of 0.01.

We evaluated the best model on the fractured and non-fractured testing sets. The model achieves a testing accuracy of 66.5%. We suspect that the relatively low accuracy of our baseline model can be attributed to the compression of our images prior to feature extraction. Most bone fractures display only minor traces in an X-ray image; successfully identifying these fractures require careful attention to details. By resizing the images so that their height and width dimensions are below 512, we may have blurred a lot of localized details in the images, and thus the feature embeddings may not effectively represent these details. This might have made it hard for the MLP to make classifications based on the embeddings.

5.2. Classification: ViT, DINOv2 small, DINOv2 large

As detailed in our methods section, alongside developing our own models, our research also explores the performance of pretrained models, specifically those using attention-based mechanisms, for bone fracture classification. We conducted experiments first using supervised learning with the 'vit-base-patch16-224' and then using self-supervised learning with both small and large configurations of DINOv2. All three models were configured similarly and fine-tuned with the same objective: to classify X-ray images into two categories—fractured and non-fractured bones. Below, we describe the setup for the DINOv2 small configuration.

To handle the data, we implemented a custom XRayDataset class in PyTorch, designed to load, preprocess, and augment the images. Each image is opened, converted to RGB format (to ensure compatibility with the model's expected input), and then transformed using DINOv2's standard preprocessing pipeline, which includes resizing and normalization. We partition our dataset into training (81%), validation (9%), and test sets (10%).

We finetuned DINOv2 using the Hugging Face transformers library. Key parameters include a learning rate of 2e-5, batch size of 16 during training and 32 during evaluation, 5 epochs, AdamW optimizer with parameters $\varepsilon = 10^{-8}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$, a weight decay of 0.01 and using cross entropy loss. To monitor training dynamics, we utilized TensorBoard, logging loss curves and accuracy metrics at every 10 steps. This granular logging helped us identify any anomalies or plateaus in the learning process

Given the critical nature of fracture diagnostics, where false negatives can lead to missed fractures, we evaluated our model using a comprehensive set of metrics:

1. Accuracy: overall correctness rate
2. Precision: Proportion of true fractures among predicted fractures
3. Recall: Proportion of correctly identified fractures (critical to minimize missed fractures)

After fine-tuning, our DINOv2-based model achieved the following performance on the test set:

- Train Loss: **0.2914**
- Test Loss: **0.3645**
- Accuracy: **0.9288**

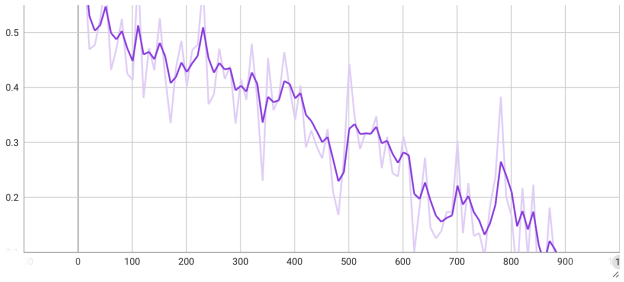

Figure 1. DINOv2 small training loss

- Precision: **0.9440**
- Recall: **0.9201**

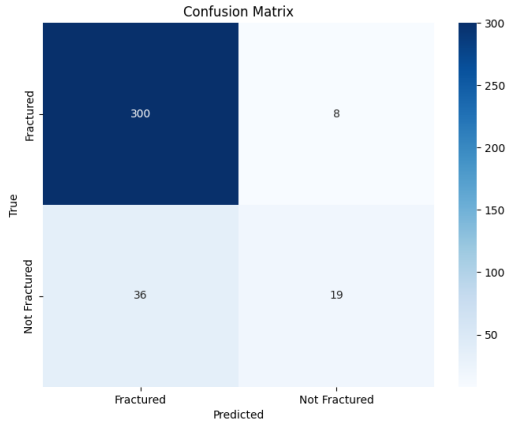

Figure 2. DINOv2 small confusion matrix

DINOv2 small correctly classifies nearly 93% of all X-ray images, demonstrating high overall accuracy. Similarly, when our model predicts that there is a fracture, it is correct 94.40% of the time. High precision is excellent for reducing the burden of false positives in medical contexts. Also, the model identifies 92.01% of all fractures. Ideally, recall should be as close to 100% as possible in medical applications to minimize the risk of missed diagnoses. Furthermore, the calculated losses indicate a slight degree of overfitting. This is likely due to the relatively small size of our dataset which restricts the diversity of training examples.

Below is a final comparison of all our models' performances.

c

Model	Test Accuracy
ResNet + MLP	66.5%
ViT	90.0%
DINOv2 small	92.8%
DINOv2 large	96.1%

Table 1. Comparison of Model Performances

5.3. Detection: Transfer Learning with YOLO

For the bone fracture detection task, we used our fractured training set of 720 X-ray images to finetune the YOLOv8 model. Most of these X-ray images containe one bone fracture, some of these images contain two, three, or four bone fractures, and one image contains up to five bone fractures. Taken together, the training set contains 922 instances of bone fractures. Prior to finetuning, we downloaded the weights of the YOLOv8 pretrained model and converted the bounding box data from Coco JSON format to YOLO text annotation format. Using the training data, we ran 100 epochs of AdamW optimization on the pretrained YOLOv8 weights and saved the model that achieves the best performance on the validation set. Then, we used the best model to detect bone fractures in the test set X-ray images. The confusion matrix of the best finetuned YOLOv8 model is as follows:

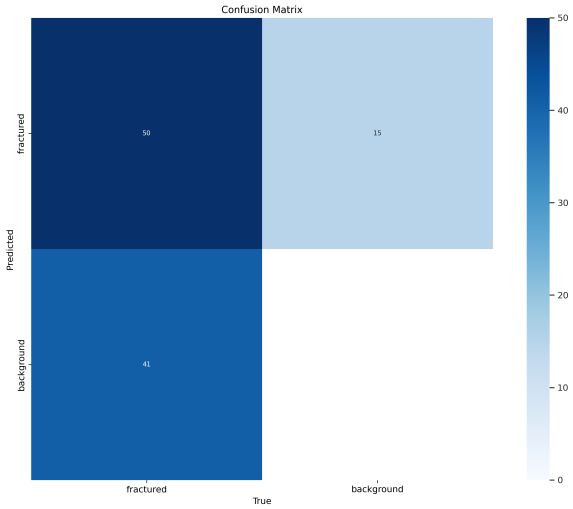

Figure 3. YOLO Confusion Matrix

In our testing set, there are 83 images with 91 instances of bone fracture. Out of these 91 bone fracture instances, 50 are correctly identified by the finetuned YOLO model, whereas the other 41 are not discovered by the finetuned model. Furthermore, the finetuned model falsely identified 15 background spots as fractures. This yields a precision of 77% and a recall of 55%.

The best finetuned model was able to achieve an mAP50 of 0.53. In other words, 53% of its predicted bounding boxes have an IoU of at least 0.5 with a ground truth bounding box. It was able to achieve an mAP50-95 of 0.22 - this is the average fraction of predicted boxes with an IoU of at least 0.50, 0.55, . . . , 0.90, 0.95 with a ground truth bounding box.

The performance of the model may not qualify for imme-

diate deployment in real-world bone fracture diagnosis applications. However, it is still significantly better than a random detector. Furthermore, while the model is only able to detect slightly more than half of all bone fracture instances, most of its detected spots do indeed contain fractures. Since bone fractures can occur in a variety of different modes, the fact that the model's precision is much higher than its recall might suggest that the model has been able to capture the characteristics of some bone fracture modes relatively well, although it still cannot capture the characteristics of other types of bone fractures.

Below we compare the ground-truth bone fracture bounding boxes with the predicted bounding boxes on one batch of testing data:

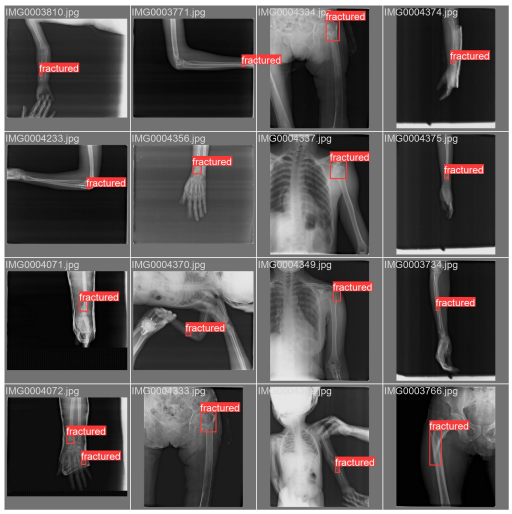

Figure 4. Ground-Truth Bounding Boxes

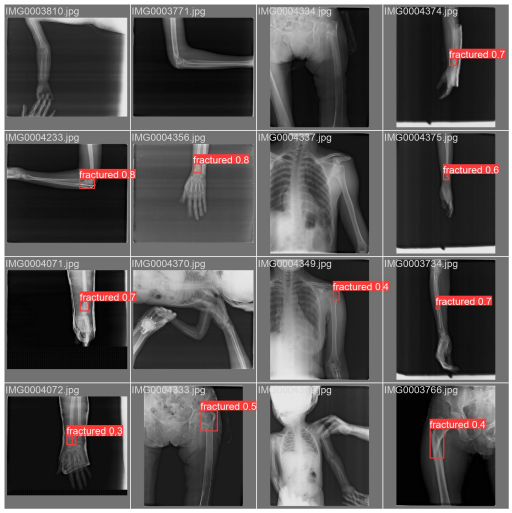

Figure 5. Predicted Bounding Boxes

As we can see, in this batch of testing data, all predicted bounding boxes of the model coincide well with ground truth bounding boxes. However, the model fails to identify a few bone fracture instances. We notice that many of the model's failure cases involve fractures that occur at thinner bone regions especially for children, as evidenced in the above figures by the two arm fractures on the left of the first row and in the third image of the fourth row. This failure mode is expected as fractures at thinner bone regions display smaller traces that are harder for YOLO's backbone convolutional filters to capture. Furthermore, the model tends to miss fractures that occur near the joints of the body. For instance, the model fails to detect the fracture at the top of the thigh and the fracture at the shoulder in the upper part of the third column. These failures may be due to the fact that both joints and fractures are displayed as skeletal discontinuities in X-ray images, and it is inherently hard to distinguish joints from fractures given their similar appearances under X-ray.

As mentioned before, most predicted bounding boxes coincide with ground truth bounding boxes, and false positives are far less common. We observed that of the few false positive cases, many involve misclassifying a natural joint as a fracture. An example of this failure mode is shown below:

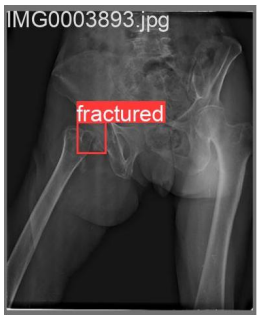

Figure 6. False Positive Example: Ground Truth Bounding Boxes

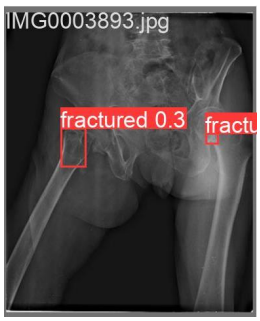

Figure 7. False Positive Example: Predicted Bounding Boxes

In this example, the model falsely treats the joint at the

top of the left thigh bone (right side of the X-ray image) as a fracture. The predicted location of the actual fracture in the image is also slightly off: while the actual fracture occurs slightly above the joint on top of the right thigh bone (left side of the image), the model predicts the joint itself as the fracture spot. Therefore, our findings suggest that skeletal joints interfere significantly with the detection of bone fractures by deep learning models.

6. Conclusion

In this study, we diverge from traditional supervised learning models by employing self-supervised learning to train our algorithms, enabling them to discern more nuanced features without extensive labeled datasets.

The DINOv2 models stood out as the highest-performing in our series of experiments, particularly in classification tasks where the DINOv2 large configuration achieved an accuracy of up to 96.1%. This model's superior performance is attributable to its sophisticated architecture developed by Meta AI. However, while DINOv2 excels in classification, it was not without challenges; the model exhibited slight overfitting and did not achieve a perfect recall. Enriching the training dataset with more varied and complex fracture cases could help improve the generalizability and robustness of DINOv2. Also, implementing more sophisticated regularization methods could help mitigate the overfitting observed.

On the other hand, the YOLOv8 model, adapted for localization tasks, achieved a mean Average Precision (mAP50) of 0.53. This indicates that while the model is generally effective at identifying the locations of fractures, there is room for improvement, especially in complex anatomical areas where fractures are small or near joints, which are challenging for the model to accurately pinpoint.

In conclusion, our research demonstrates significant potential for using advanced machine learning techniques in the field of radiology. By further developing these models, there is a promising path toward fully automated, highly accurate diagnostic tools that can substantially benefit clinical practices by reducing the workload on radiologists and improving diagnostic outcomes for patients.

References

[1] I. Abedeen et al. Fracatlas: A dataset for fracture classification, localization and segmentation of musculoskeletal radiographs. volume 10, page 521, 2023. 2

[2] I. M. Baltruschat, H. Nickisch, M. Grass, T. Knopp, and A. Saalbach. Comparison of deep learning approaches for multi-label chest x-ray classification. *Scientific reports*, 9(1):6381, 2019. 1

[3] P. Bharati and A. Pramanik. Deep learning techniques—r-cnn to mask r-cnn: a survey. *Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2019*, pages 657–668, 2020. 4

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 1, 3

[5] B. Guan, J. Yao, S. Wang, G. Zhang, Y. Zhang, X. Wang, and M. Wang. Automatic detection and localization of thighbone fractures in x-ray based on improved deep learning method. *Computer Vision and Image Understanding*, 216:103345, 2022. 2

[6] F. Hardalaç, F. Uysal, O. Peker, M. Çiçeklidağ, T. Tolunay, N. Tokgöz, U. Kutbay, B. Demirciler, and F. Mert. Fracture detection in wrist x-ray images using deep learning-based object detection models. *Sensors*, 22(3):1285, 2022. 2

[7] N. Jaccard, T. W. Rogers, E. J. Morton, and L. D. Griffin. Detection of concealed cars in complex cargo x-ray imagery using deep learning. *Journal of X-ray Science and Technology*, 25(3):323–339, 2017. 2

[8] S. H. Kong, J.-W. Lee, B. U. Bae, J. K. Sung, K. H. Jung, J. H. Kim, and C. S. Shin. Development of a spine x-ray-based fracture prediction model using a deep learning algorithm. *Endocrinology and Metabolism*, 37(4):674, 2022. 2

[9] Y. Li, Z. Huang, X. Dong, W. Liang, H. Xue, L. Zhang, Y. Zhang, and Z. Deng. Forensic age estimation for pelvic x-ray images using deep learning. *European radiology*, 29:2322–2329, 2019. 1

[10] C. Liu et al. Object detection based on yolo network. In *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, 2018. 2

[11] H. P. Nguyen, T. P. Hoang, and H. H. Nguyen. A deep learning based fracture detection in arm bone x-ray images. In *2021 international conference on multimedia analysis and pattern recognition (MAPR)*, pages 1–6. IEEE, 2021. 2

[12] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. 4

[13] M. E. Sahin. Image processing and machine learning-based bone fracture detection and classification using x-ray images. *International Journal of Imaging Systems and Technology*, 33(3):853–865, 2023. 2

[14] S. Targ, D. Almeida, and K. Lyman. Resnet in resnet: Generalizing residual architectures. *arXiv preprint arXiv:1603.08029*, 2016. 1, 3

[15] D. Yadav and S. Rathor. Bone fracture detection and classification using deep learning approach. In *2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC)*, pages 282–285. IEEE, 2020. 2

[16] S. Yang, J. Kweon, J.-H. Roh, J.-H. Lee, H. Kang, L.-J. Park, D. J. Kim, H. Yang, J. Hur, D.-Y. Kang, et al. Deep learning segmentation of major vessels in x-ray coronary angiography. *Scientific reports*, 9(1):16897, 2019. 1