Representation Learning for Bone Fractures

AI For Healthcare - Ex3

Using Architectures DenseNet169 & SimCLR (ResNet-50)

Shir Nitzan, Timor Baruch, Hadar Pur

July 21, 2023

1 Overview

This document provides a comprehensive walkthrough of the process we undertook in our course assignment. The primary goal of our study was to classify bone fractures in X-ray images from the MURA dataset [1] using Self-Supervised Learning (SSL) techniques and compare its performance with that of a Baseline model. Our assignment focused particularly on three bones: the elbow, hand, and shoulder. The assignment encompassed several steps:

1. The initial phase involved exploring the MURA dataset, establishing a naive baseline, analyzing the distribution of data, and visualizing and evaluating the results, including both successful and unsuccessful predictions.

2. The subsequent step involved utilizing a representation learning approach. For this purpose, we chose SimCLR, employing a ResNet-50 model. We performed an analysis, comparing the performance of the model when using varying amounts (1%, 10%, and 100%) of labeled data. This method aligns with the approach suggested by Zhai et al[2].

This report details our procedure, which encompasses data exploration, preprocessing, model application, and the evaluation of results. We include the associated Python code to illustrate the methodologies implemented during the assignment.

2 Mura Dataset

The core of our study is the MURA dataset[1], a substantial collection of musculoskeletal radiographs. The dataset encompasses 14,863 studies from 12,173 patients, aggregating to a total of 40,561 multi-view radiographic images. These images are segregated into seven types of upper extremity radiographic studies: elbow, finger, forearm, hand, humerus, shoulder, and wrist. Each study within this dataset was meticulously labeled as normal or abnormal by board-certified radiologists from Stanford Hospital.

[1] MURA dataset

In the scope of this assignment, we specifically focused on three types of studies: elbow, hand, and shoulder. During the data exploration phase, we transformed the selected subset of the MURA dataset into a suitable format for our machine learning models. We performed normalization and applied various data augmentation techniques, such as flipping, rotating, and translating, to enhance the robustness and generalization of our models.

Subsequently, the prepared dataset was used for training, and split into different subsets based on the quantity of labeled data - specifically, 1%, 10%, and 100%. These subsets allowed us to methodically evaluate and understand the performance of our models based on the varying amount of labeled data used. This approach offered crucial insights into the effectiveness of representation learning, specifically SimCLR, in the domain of medical image classification, while also giving us a deeper understanding of how the quantity of labeled data influences model performance.

3 Data Exploration

The MURA dataset consists of two main folders: the train set and the validation set. Each of these folders contains subfolders representing different body parts, such as the elbow, finger, forearm, hand, humerus, shoulder, and wrist.

Within each body part subfolder, there are folders corresponding to individual patients. These patient folders are uniquely identified by their assigned numbers along with the corresponding labels indicating whether the image is positive (indicating the presence of a bone fracture) or negative (indicating the absence of a fracture). Each patient folder contains a collection of X-ray images of the specific body part.

As a part of our data exploration, we have broken down the composition of both the training and validation datasets in terms of the count of normal and abnormal instances for each of the three study types: elbow, hand, and shoulder. This breakdown is summarized in the following tables:

Part	**Normal**	**Abnormal**	**Total**
Elbow	2925	2006	4931
Hand	4059	1484	5543
Shoulder	4211	4168	8379

Table 1: Training Dataset

Part	**Normal**	**Abnormal**	**Total**
Elbow	325	230	465
Hand	271	189	460
Shoulder	285	278	563

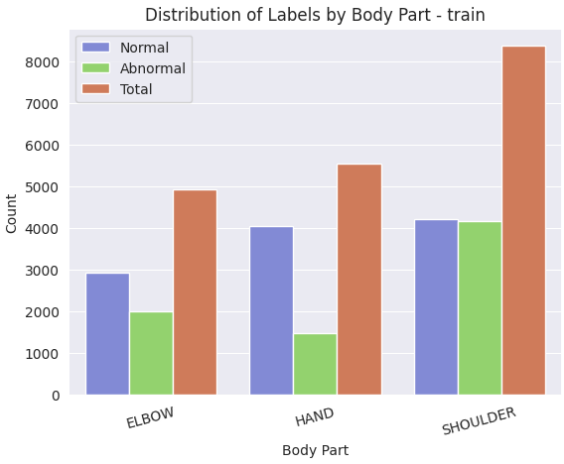
Table 2: Validation Dataset

Analyzing the tables, it's clear that the dataset isn't balanced, with the number of normal cases exceeding the number of abnormal cases for all three body parts in the training dataset. The hand studies have the most significant imbalance with around 73.3% normal cases, followed by elbow studies with 59.3% normal cases. For shoulder studies, the distribution is almost equal, with 50.3% normal cases.

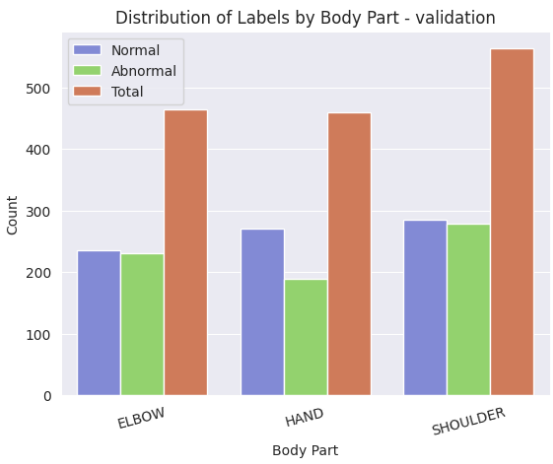
In the validation dataset, the imbalance is less significant. The shoulder studies show a near-equal distribution of normal and abnormal cases, while elbow and hand studies have a slightly higher number of normal cases.

An analysis of the data reveals an imbalance in the number of normal and abnormal cases for each study type. This is a common challenge in many medical datasets and presents an interesting avenue for further research and refinement of the model. However, in the scope of this assignment, we did not explicitly address this issue. It's crucial to be aware that this imbalance could potentially introduce biases in our model's learning process, but assessing and mitigating such biases will require further investigation beyond the current work.

Examining the images within the patient folders, we observed that each folder typically contains between 1 to 5 images of the corresponding body part. This variability in the number of images per patient contributes to the diversity and richness of the dataset, allowing for a comprehensive exploration of bone fractures across different patients and imaging instances. Our main focus was on the hand, elbow, and shoulder radiographic study types from the MURA dataset.



Training Dataset



Validation Dataset

Figure 1: Bar char of the data distribution according to labels (normal/abnormal) and classes

4 Inspection of Representative Images

To better understand the nature of our data, we inspected a selection of images from the three study types: elbow, hand, and shoulder. This inspection involved viewing both normal and abnormal X-ray images from each category, providing us with a valuable insight into the variations and complexities inherent in our dataset.

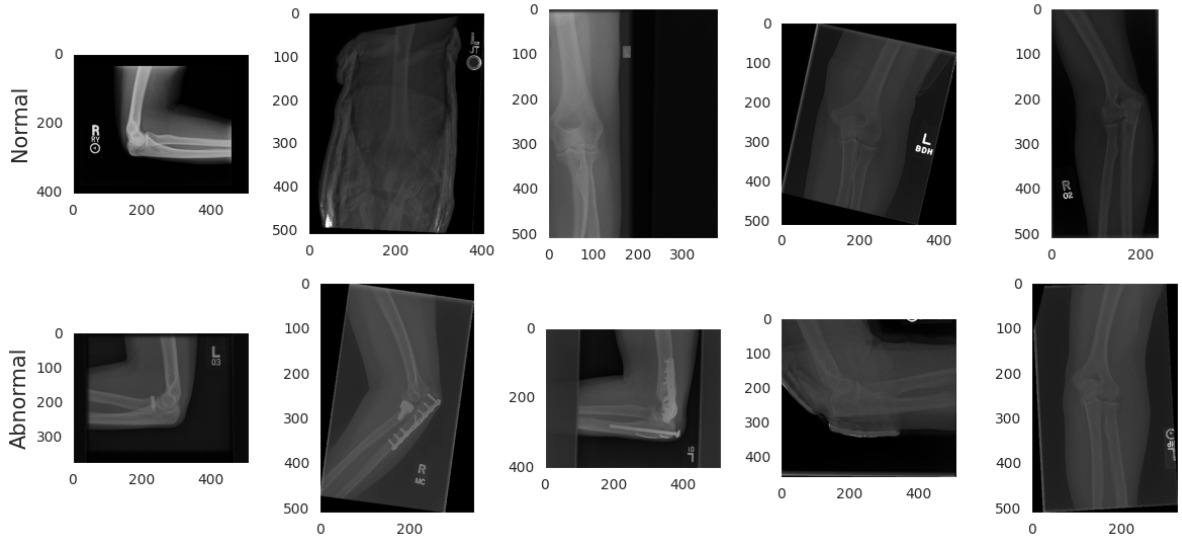


Figure 2: Elbow dataset - A selection of normal and abnormal X-ray images

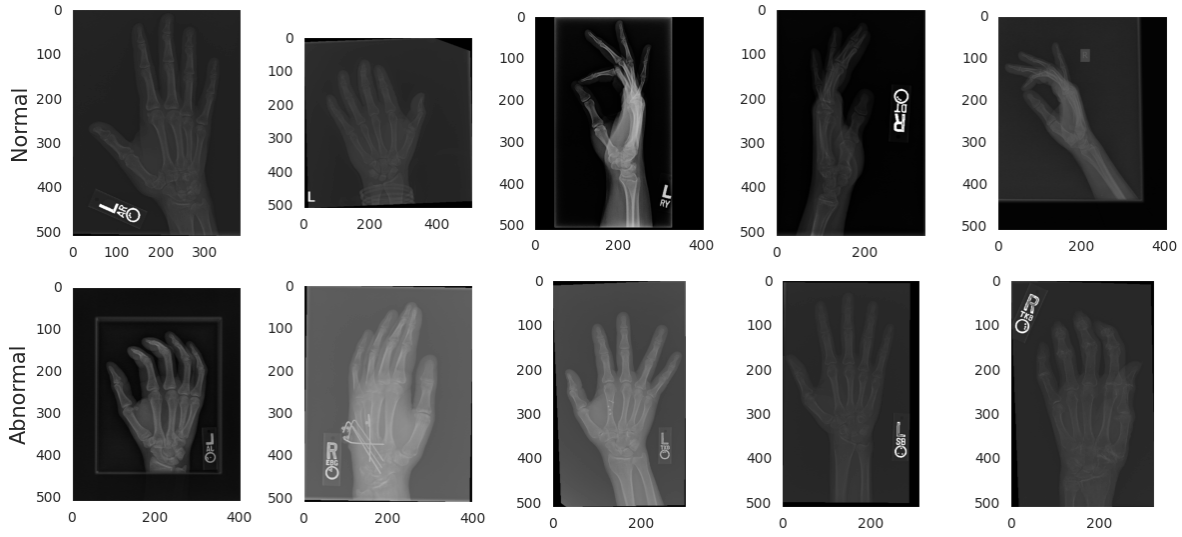


Figure 3: Hand dataset - A selection of normal and abnormal X-ray images

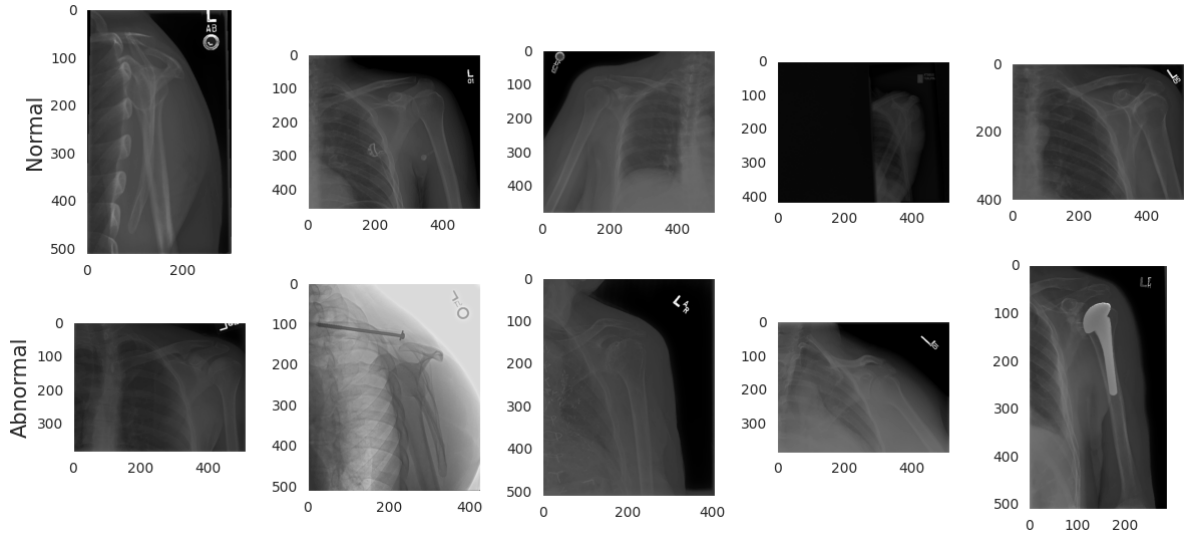


Figure 4: Shoulder dataset - A selection of normal and abnormal X-ray images

Each of these figures comprises eight X-ray images, half of which depict normal cases and the other half represent abnormal cases. By observing these images, we noted that the X-ray images display significant variations in terms of positioning, contrast, and brightness. These factors underscore the challenges posed in our classification task, as our model must navigate these complexities to deliver accurate results.

5 Data Preprocessing

During the data preprocessing phase, several actions were executed to refine our dataset for subsequent training and evaluation. Initially, the original training set was randomly partitioned into an 80% subset for model training and a 20% subset for testing. This separation ensured having a separate dataset to evaluate the model's performance after training.

In order to ensure consistency in our workflow and evaluation, we modified the original MURA dataset's validation set. Specifically, we renamed and repurposed the validation set as our testing set, thus delineating our data into three distinct categories: training, validation, and testing.

By repurposing the validation set as the testing set, we were able to assess the model's performance on previously unseen and unfamiliar data. This strategy offered a robust measure of the model's generalization capabilities, thereby bolstering the credibility of our evaluation.

Part	**Train**	**Validation**	**Test**
Elbow	3964	967	465
Hand	4415	1128	460
Shoulder	6723	1656	563

Table 3: Dataset split for Training, Validation, and Test for both Baseline and SSL models

6 Data Augmentation and Normalization

Data augmentation and normalization form a critical part of our preprocessing strategy, enhancing the generalization capability of our model and mitigating overfitting.

6.1 Data Augmentation

During the data augmentation phase, we introduced random modifications to our training data, effectively enriching our dataset and pushing the model to learn from more diverse and generalized features. Specifically, we implemented operations such as flipping images horizontally and rotating images within a predefined range, alongside optional transformations including scaling and translating.

6.2 Data Normalization

Subsequent to augmentation, we performed data normalization. This essential step ensures that each input parameter (pixel, in this case) has a comparable data distribution, enabling faster and more efficient convergence during training.

7 Model Architecture

7.1 Baseline - DenseNet169

The DenseNet-169 architecture serves as the fundamental model for bone fracture X-ray classification in this task [3]. As a powerful deep learning architecture, DenseNet-169 has demonstrated exceptional performance in various image classification tasks. Its distinctive dense connectivity pattern enables each layer to connect with all others in a feed-forward manner, promoting efficient feature reuse and gradient flow. This design empowers the model to learn intricate patterns and representations from the input X-ray images effectively.

With its substantial depth and parameter efficiency, DenseNet-169 provides a robust foundation for our bone fracture classification task. It serves as a starting point for training and allows for meaningful performance comparisons with other approaches. We assisted the implementation provided in the official Git repository of DenseNet on MURA dataset[2].

7.2 SimCLR - ResNet50

SimCLR is widely recognized as a powerful approach for self-supervised learning, designed to learn meaningful representations from unlabeled data using contrastive learning. In the SSL section of this task for bone fracture X-ray classification, we employ the SimCLR framework with a ResNet-50 implementation.

During training, SimCLR distinguishes between similar and dissimilar pairs of augmented versions of the same image. By maximizing the agreement between representations of similar image views and minimiz-

[2] DenseNet on MURA dataset

ing the agreement between representations of dissimilar views, the model learns robust and discriminative features. This process tailors ResNet-50 specifically for bone fracture classification.

The primary advantage of using SimCLR is its ability to leverage the large quantity of unlabeled data from the MURA dataset during the pretraining phase. It learns generalizable representations from the unlabeled data, which provides a solid foundation for subsequent learning. In the fine-tuning phase, the model is further trained on a specified percentage of labeled data (1%, 10%, or 100%), enabling it to adapt its learning to the specific task of bone fracture classification. To implement SimCLR, we utilized the official implementation provided by Google in their Git repository[3].

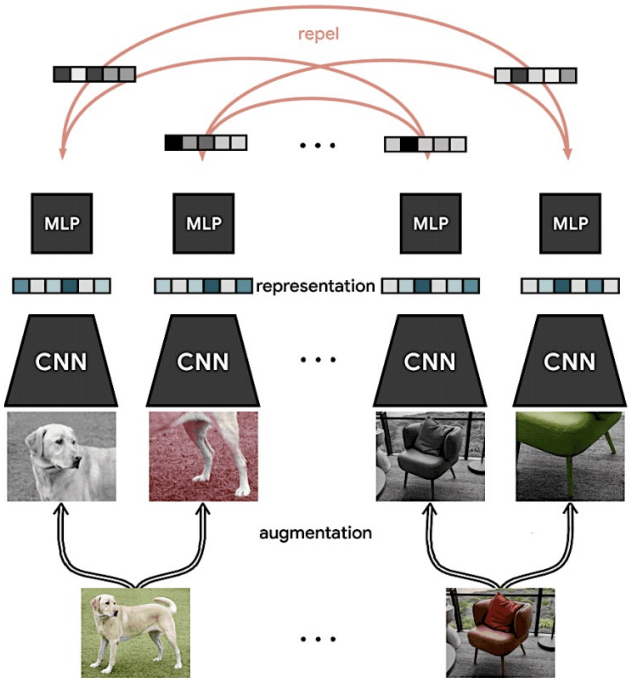


Figure 5: SimCLR framework uses CNN and MLP layers to generate similar projections for augmented versions of an image and dissimilar projections for different images, even within the same object class. It excels in recognizing image transformations and learning representations for similar concepts (e.g., chairs vs. dogs). These representations can be fine-tuned and used for classification tasks.[4]

8 Hyperparameter Tuning

In our implementation, we followed the default parameter settings provided in the SimCLR framework's Git repository for both fine-tuning and pre-training stages. These default parameters were carefully chosen and widely used in SimCLR-based experiments, ensuring consistency and allowing for fair comparisons with other works.

[3] SimCLR official model repository
[4] Google Research official blog

While we made some minor adjustments, such as reducing the evaluation batch size and training size to suit our specific requirements and set of computational resources, we maintained the core parameter settings as recommended by the SimCLR framework. This approach ensured that we were utilizing the framework's default values for crucial parameters such as the learning rate, temperature and optimizer. Moreover, we opted to use a smaller variant of the ResNet architecture, 50 layers instead of 101, and reduced number of epochs, due to limited computational resources and GPU capabilities.

In the baseline model, we maintained a training duration of 15 epochs, a learning rate of 2e-5, and utilized a DenseNet with a depth of 169. The batch size during training was set to 8, while for evaluation and testing, a batch size of 16 was used.

8.1 Learning rate

In our experiments, we explored the impact of different learning rates (lr) on the model's performance during the fine-tuning phase for the SSL model. We conducted experiments using lr values of 0.3 and 0.1.

The learning rate is a critical hyperparameter as it determines the step size for updates to the model's parameters during training. Initially, a higher learning rate, such as 0.3, facilitates faster convergence and large parameter updates. This is beneficial in the early stages of fine-tuning when the model needs to rapidly adapt to the bone fracture classification task.

However, as training progresses, it becomes advantageous to reduce the learning rate, for instance, to 0.1. A lower learning rate allows the model to make finer adjustments to its parameters, which can help prevent overfitting and improve the model's generalization on the labeled data. This reduction in the learning rate is often known as learning rate decay, and it aids in better convergence by allowing the model to navigate the parameter space with more precision and find optimal points in the loss function.

In simple terms, the learning rate decay strategy encourages the model to make big adjustments in the beginning to learn fast and later make smaller, more precise adjustments to fine-tune its representations for the bone fracture classification task. Detailed experiments with lr values of 0.1 and 0.3 are provided within the SSL notebook.

In the baseline model, we experimented with various learning rates during training and found that a learning rate of 2e-5 yielded the best results. This lower learning rate allowed for smaller weight updates, which can facilitate convergence and prevent the learning process from diverging. Our decision aligns with recommendations from the official Git repository, making it appropriate for our bone fracture classification task.

8.2 Epochs

Epochs played a crucial role in the model training process. In the baseline approach, we observed that training for 15 epochs yielded good enough results, as further epochs did not lead to significant improvements and even caused overfitting in the shoulder class. However, in the SSL phase with the more complex model, we

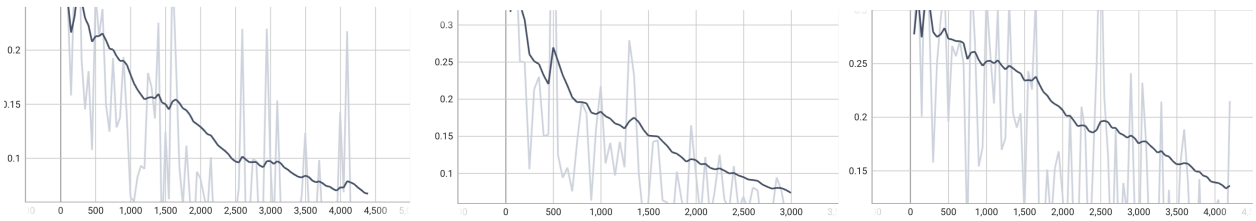
used 50 epochs for pre-training and witnessed notable performance gains. If we had more GPU and memory resources, we could further increase the number of epochs and consider enlarging the ResNet depth since it appeared to continue improving.

In the SSL fine-tuning phase, we chose to use 20 epochs as the number of training iterations. This decision was based on the fact that the model had already undergone pre-training and acquired meaningful representations from the unlabeled data. The primary focus of fine-tuning was to adapt the model to the bone fracture classification task using a limited number of epochs, while still benefiting from the valuable pre-trained features. By doing so, we aimed to achieve optimal performance and enhance the model's ability to accurately classify bone fractures.

9 Baseline (DenseNet-169)

9.1 Train

We conducted experiments to enhance the baseline approach for bone fracture classification using DenseNet-169. Our experiments explored techniques such as data augmentation, data normalization, and optimization strategies. These efforts aimed to improve the model's performance and accuracy, guiding us towards refined approaches for bone fracture classification. In our experimentation, we explored different combinations of epochs and batch sizes to optimize the model's performance while considering resource constraints. After thorough evaluation, we found that training the model for 15 epochs for each class yielded satisfactory results. Beyond this point, further epochs did not lead to significant improvements in performance. Additionally, due to limitations in GPU resources, we determined that a batch size of 16 for the train set and 16 for tests and validation sets provided a balance between computational efficiency and model convergence.



Elbow Hand Shoulder

Figure 6: Baseline; Training loss over labeled train dataset

9.2 Evaluation

In evaluating the baseline model, our primary objective is to assess its performance and effectiveness. We focus on key metrics, including accuracy, precision, and recall, to evaluate the model's performance on specific testing subsets. These evaluations provide us with valuable insights into the model's generalization capabilities, allowing us to understand how well it performs on unseen data. We assess the performance of

the model on each relevant class, namely hand, shoulder, and elbow. This allows us to analyze the model's effectiveness in classifying bone fractures specific to these body parts. The results are shown in Table 4.

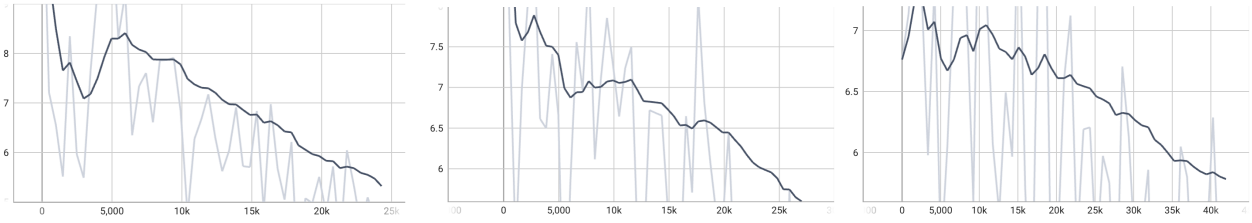
10 SSL - SimCLR (ResNet-50)

10.1 Pre-training

In the pre-training phase, our goal was to take advantage of the abundance of unlabeled data in the MURA dataset. We used the SimCLR framework to train a ResNet-50 model, which does not make use of the labels but instead learns representations based on SSL methods.

During this stage, the model learned by trying to differentiate between augmented versions of the same image (positive pairs) and different images (negative pairs). The objective was to maximize agreement among positive pairs while minimizing agreement among negative pairs, pushing the model to understand complex patterns and structures from the images.

The pre-training process was a long one, involving 50 training epochs. The reason behind this number is tied to the objective of the pre-training phase, which is to learn as much as possible from the unlabeled data. SimCLR, as a self-supervised learning technique, requires sufficient training time to learn meaningful representations. As we increase the number of epochs for the pre-training process, the representations that SimCLR learns become more nuanced and accurate, contributing to better downstream task performance [4]. After some experimentation with different numbers of epochs, we found that a lower number of epochs led to sub-optimal representations, thereby impacts the model's performance in the fine-tuning stage.



Elbow Hand Shoulder

Figure 7: SSL; Training loss over labeled train dataset

10.2 Fine-tuning

In the fine-tuning stage, the pre-trained model was further trained on a smaller, labeled subset of the dataset. This process helped the model adjust the representations learned in the pre-training phase to more specific tasks of classifying the medical images into normal or abnormal.

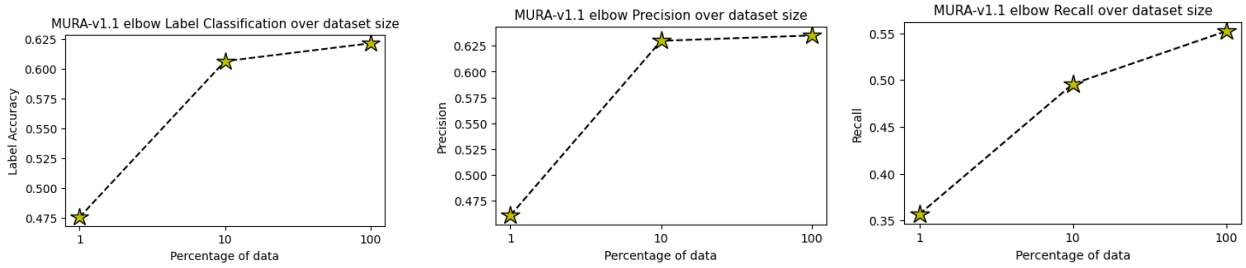
Our fine-tuning was an iterative process involving various proportions of labeled data, starting from 1%, and moving on to 10%, and finally, 100%. This technique allowed us to assess the impact of the amount of labeled data on the model's performance.

During the fine-tuning stage, we used a small number of training epochs, usually around 10 or 20. This decision was based on two reasons. First, the pre-trained model from SimCLR provided a good starting point, so extensive fine-tuning was not necessary. Second, our experiments revealed that adding more epochs did not significantly improve performance in the fine-tuning phase.

This two-stage approach of long pre-training on unlabeled data followed by short fine-tuning on labeled data allowed us to fully utilize our available resources. We were able to learn robust representations from the much larger unlabeled dataset, which proved valuable when fine-tuning the model with the smaller labeled dataset.

10.3 Elbow Dataset Results

The MURA elbow dataset comprised of 465 test examples, 3,964 train examples, and 967 validation examples. When utilizing 10% of labeled data, we used 385 train examples, and when utilizing 1% of labeled data, we used 36 train examples.

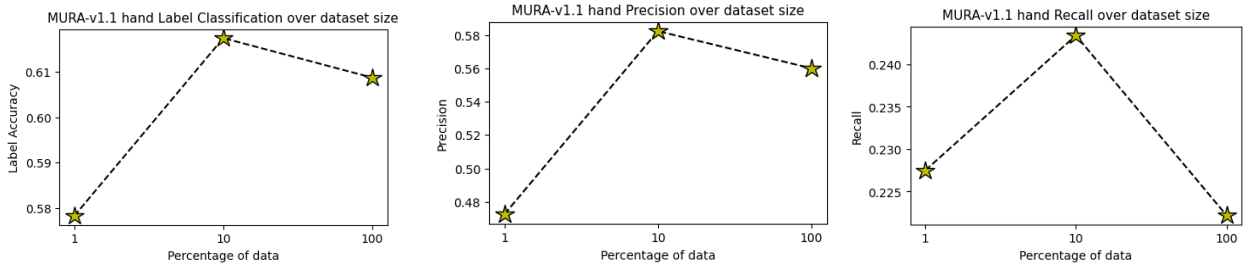


Accuracy Precision Recall

Figure 8: Label accuracy, precision and recall comparison over different fractions of elbow labeled data

10.4 Hand Dataset Results

The MURA hand dataset comprised of 460 test examples, 4,415 train examples, and 1,128 validation examples. When utilizing 10% of labeled data, we used 433 train examples, and when utilizing 1% of labeled data, we used 44 train examples.

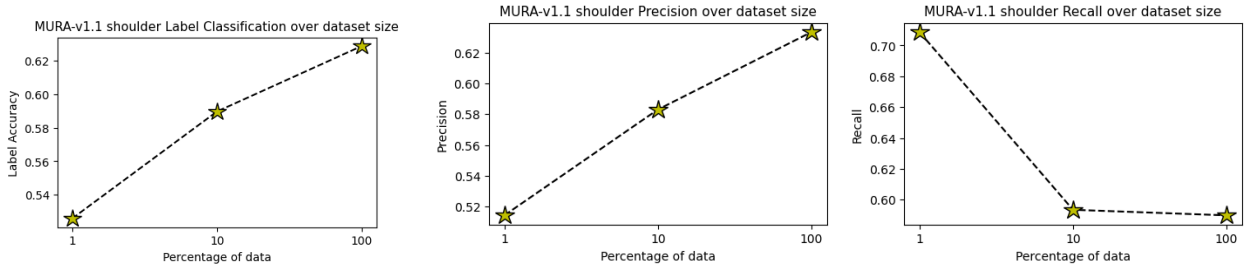


Accuracy Precision Recall

Figure 9: Label accuracy, precision and recall comparison over different fractions of hand labeled data

10.5 Shoulder Dataset Results

The MURA shoulder dataset comprised of 563 test examples, 6,723 train examples, and 1,656 validation examples. When utilizing 10% of labeled data, we used 722 train examples, and when utilizing 1% of labeled data, we used 65 train examples.



Accuracy Precision Recall

Figure 10: Label accuracy, precision and recall comparison over different fractions of shoulder labeled data

10.6 Comparison of Experiments

This table presents the comparison of experiments between the baseline and SSL on three different body parts: Elbow, Hand, and Shoulder. The comparison was performed between the evaluation of the baseline model after training and the evaluation step during the fine-tuning of the SSL model, which occurred after pre-training. The experiments involve different percentages of fine-tuning on the labeled data (1%, 10%, and 100%). The key metrics evaluated are Label Accuracy, Precision, and Recall.

Exp	Body Part	Learning rate	Epochs for pre-train + train	Accuracy	Precision	Recall
Baseline	Elbow	2e-5	15	**0.822**	0.506	0.333
Fintune 1pct	Elbow	0.1	50+20	0.475	0.460	0.356
Fintune 10pct	Elbow	0.1	50+20	0.606	0.629	0.495
Fintune 100pct	Elbow	0.1	50+20	0.621	**0.635**	**0.552**
Baseline	Hand	2e-5	15	**0.784**	0.479	0.225
Fintune 1pct	Hand	0.1	50+20	0.578	0.472	0.227
Fintune 10pct	Hand	0.1	50+20	0.617	**0.582**	**0.243**
Fintune 100pct	Hand	0.1	50+20	0.608	0.560	0.222
Baseline	Shoulder	2e-5	15	**0.778**	0.540	0.475
Fintune 1pct	Shoulder	0.1	50+20	0.525	0.514	**0.708**
Fintune 10pct	Shoulder	0.1	50+20	0.589	0.583	0.593
Fintune 100pct	Shoulder	0.1	50+20	0.628	**0.633**	0.589

Table 4: Comparison of Experiments between baseline and SSL

10.7 Results

Our experimental results shed light on the performance of the Self-Supervised Learning approach using SimCLR for bone fracture classification. We conducted the experiments on three distinct body parts: Elbow, Hand, and Shoulder, and evaluated the models based on label accuracy, precision, and recall metrics.

In our analysis, we also compared the SimCLR model, which utilized the ResNet-50 architecture, to the baseline model consisting of DenseNet-169. The comparison involved evaluating the baseline model and SSL models using 100%, 10%, and 1% of the labeled data.

10.7.1 Elbow Dataset

In the Elbow dataset, the baseline model achieved the highest accuracy of 82%, surpassing the best accuracy of the SSL experiments using all labeled data, which reached 62% label accuracy rate. However, when fine-tuning the model with 100% labeled data after initial training with SimCLR, better precision (63%) and recall (55%) scores were achieved. Despite the relatively lower accuracy, this model performed well in terms of precision and recall, highlighting the advantages of representation learning.

10.7.2 Hand Dataset

In the case of the Hand dataset, the baseline model demonstrated the highest accuracy at 78% over test data. However, when utilizing only 10% of the labels fine-tuned after representation learning with SimCLR, we observed the best precision and recall rates of 58% and 24%, respectively. This indicates that even with a small portion of labeled data, representation learning can still yield good results.

It is worth noting that the dataset had a significantly lower proportion of abnormal data compared to normal data (approximately four times smaller). This imbalance in the dataset may have influenced the limited improvement in recall rates, even with more labeled data included.

The dataset's imbalance may explain why better results were obtained with 10% labeled data compared to 100%. In imbalanced datasets, the model can face difficulties learning from the minority class, leading to lower recall rates. However, with 10% labeled data, the imbalance is less pronounced, enabling the model to focus on both normal and abnormal samples, leading to improved recall, precision and label accuracy rates.

10.7.3 Shoulder Dataset

In the Shoulder dataset, the baseline model achieved the highest accuracy of 78%, outperforming the SSL approach which achieved a maximum accuracy of 62% over all labeled data. However, when fine-tuning the model with 100% labeled data after SimCLR representation learning, a strong balance between precision (63%) and recall (59%) was achieved. Surprisingly, even with only 1% labeled data, the SSL approach demonstrated impressive recall rates of 71%.

The lower recall rates for 100% labeled data can be attributed to potential challenges introduced by a larger and more specific set of labeled examples, including mislabeled or noisy samples. Additionally, the model

may encounter a wider range of variations and complexities within the data, making it harder to distinguish between normal and abnormal cases accurately.

As a result, the model's recall rate decreases to 58% for 100% labeled data, indicating that it may be missing a significant number of true positive cases due to potential noise or inconsistencies in the labeled data. In contrast, using a smaller labeled dataset, such as 10%, may be reducing the dataset imbalance and allows the model to focus on both normal and abnormal samples, resulting in better recall rate.

11 Conclusion

The outcomes of our experiments underscore the power of representation learning, specifically SimCLR, in leveraging small amounts of labeled data for fine-tuning. While the baseline models exhibit superior accuracy, the models using representation learning demonstrate improved precision and recall, implying a more balanced performance. The fine-tuning level offering the best balance between these metrics is variable and depends on the body part under examination. Further research can explore fine-tuning strategies to optimize performance metrics for different body parts.

12 Discussion

In this study, we explored the effectiveness of the SimCLR framework for bone fracture X-ray classification using the MURA dataset. Our experiments included both a baseline approach with DenseNet-169 and a SSL approach with SimCLR using ResNet-50. The results revealed that the baseline models achieved high accuracies, showcasing their effectiveness in classifying bone fractures. However, the SSL approach showed promise by leveraging the power of representation learning with unlabeled data. Notably, even with a small percentage of labeled data, the SSL approach demonstrated impressive accuracy rates over test set, emphasizing the ability of representation learning to effectively utilize limited labeled datasets. Additionally, fine-tuning the SSL models with 100% labeled data further improved precision and recall - compare to the baseline, indicating the importance of incorporating more specific labeled information. These findings highlight the potential of SSL and representation learning techniques in enhancing bone fracture classification models and encourage further exploration in this domain.

13 Code

Baseline - You can access our public Colab notebook here

SSL - You can access our public Colab notebook here

References

[1] Pranav Rajpurkar et al. "Mura: Large dataset for abnormality detection in musculoskeletal radiographs". In: *arXiv preprint arXiv:1712.06957* (2017).

[2] Xiaohua Zhai et al. "S^4L: Self-Supervised Semi-Supervised Learning". In: *Conference on Neural Information Processing Systems*. 2020. URL: https://arxiv.org/pdf/2006.10029.pdf.

[3] P. Rajpurkar et al. "MURA Dataset: Towards Radiologist-Level Abnormality Detection in Musculoskeletal Radiographs". In: *ArXiv e-prints* (Dec. 2017). arXiv: 1712.06957 [physics.med-ph].

[4] Ting Chen et al. *A Simple Framework for Contrastive Learning of Visual Representations*. 2020. DOI: 10.48550/ARXIV.2002.05709. URL: https://arxiv.org/abs/2002.05709.