Deep Learning to Automate Identification and Characterization of Rib Fractures on Chest Computed Tomography Scans

Jeff Choi MD MSc
Stanford University
jc2226@stanford.edu
& *SathyaEdamadaka*
Stanford University
sath1@stanford.edu
& *DavidBrown*
Stanford University
davidwb@stanford.edu

Abstract

Rib fractures are among the most common traumatic injuries in the United States. Characterizing detailed rib fractures are critical to inform clinical decisions, yet in-depth characterization is rarely performed due to the manual burden required for radiologists to annotate these injuries on computed tomography scans. We aimed to develop a deep learning model that can automate identification and characterization of rib fractures on chest computed tomography scans and guide real-time clinical decision-making.

Our dataset comprised 5,000 annotated rib fractures from 660 chest computed tomography scans within the publicly-available RibFrac dataset. Each image (slice CT scan) was labeled by radiologists as containing no, undisplaced, displaced, segmental (2 fractures in one bone), or buckle (partly displaced) fractures. We trained a 2D convolutional UNet model that was trained using loss functions that combine cross-entropy loss and DICE scores, and weighted classes to account for data imbalance. We also calculated the percent displacement for each rib fracture by using a Breadth First Search algorithm and linear regression to compute angular offset between two bones.

We achieved binary DICE score of 0.88 and macro F1 of 0.26 on our validation set for discriminating rib fractures. However, our model had challenges classifying rib fractures into specific subtypes. In the absence of ground truth labels for percent rib displacement, we qualitatively confirmed that our model also accurately quantifies percent displacement for fractures with less than and greater than 100% displacement.

1. Introduction

Traumatic rib fractures are common injuries that affect over 500,000 Americans annually.[1] Identifying the number of rib fractures and their specific injury characteristics (e.g. degree of displacement) are critical for major clinical decisions, from admitting patients for inpatient care to performing an operation to fix the rib fractures. [2, 3] Moreover, many observational studies exploring the association between injuries and an outcome (e.g. mortality, hospital length of stay) adjust for the number of rib fractures as a confounding variable. Identifying and characterizing rib fractures is critical for both clinical decision-making and trauma research.

Unfortunately, enumerating and characterizing rib fractures requires intensive, manual computed tomography (CT) scan review. This task is especially impractical for the many patients who suffer multiple injuries beyond rib fractures, all of which require prompt radiology review. Despite recent consensus guidelines on characterizing rib fractures (e.g. by degree of displacement, type of fracture pattern, location), few radiology departments adopted these guideline, and many only report the total number of rib fractures in a CT scan. Existing nationwide and institutional databases classify rib fractures binarily (yes or no rib fractures); the lack of granular rib fracture characteristics has been an impediment toward delivering the best evidence-based care possible.

2. Related Work

To our knowledge, only two prior studies have attempted to segment rib fractures on CT scans. [4, 5] Yao et al evaluated 1707 CT scans of patients with rib fractures and fol-

lowed three general steps: 1) segmented bones using U-Net (not modified); 2) remove non-rib bones (e.g. scapulae, sternum, vertebrae); 3) classified ribs using 3D DenseNet. Their dataset comprised CT scans (¡2mm per slice) from a single institution, labeled by five radiologists.

U-Net is a convolutional neural network developed for biomedical image segmentation that comprises an encoder (stack of convolutional and max pooling layers) and corresponding decoder paths. [6] The architecture facilitates semantic segmentation and allows a final 64-component feature vector to be mapped to the desired number of classes, facilitating transfer learning.

Rib fracture classification (binary: present or absent) was performed by adapting DenseNet [7] architecture into a 3 dimensions. DenseNet was developed to address the problem of vanishing gradients associated with deeper convolutional neural networks. DenseNet connects narrow layers directly with one another in a feed-forward fashion, where each layer accepts feature maps from all preceding layers as inputs. This ability for every layer to access gradients from the loss function and the input image alleviates the vanishing gradient problem and reduces the number of learnable parameters.

They evaluated model performance by comparing rib fractures identified by their model with fractures identified by radiologists, and achieved 0.890 F1 score.

The decision to remove non-rib bones was a strong study design choice; since the authors used a vanilla U-Net model on a dataset with relatively limited sample size, segmenting ribs from other bones within the chest CT scans may have been challenging. Based on the fact that non-rib bones have characteristic shapes common across every human, the authors simply removed these bones in the pre-processing step. This likely saved effort needed in modifying the U-Net architecture to specifically segment ribs from the CT scan. Another strength of Yao et al's study was using focal loss during training. [8] Focal loss dynamically applies a modulating term to the usual cross entropy loss to emphasize learning difficult misclassified examples as opposed to easy examples.

The study had several limitations. First, the dataset comprised images from a single institution (i.e. not externally validated) and the need to remove non-rib bones as part of pre-processing could hinder model transferability. Second, the test set only comprised 100 scans, which is quite small. Third, although the model detected rib fractures well, it did not characterize them according to aforementioned consensus characterization taxonomy.

Similar to Yao et al, Jin et al similarly adapted U-Net to identify rib fractures using 900 chest CT scans (total of 7,473 annotated rib fractures) from a single institution: their algorithm was termed "FracNet". Input images comprised 1-1.25mm thick CT slices, and were annotated by 5 radiologists, who labeled images binarily (rib fracture vs no rib fracture).

The FracNet model comprised a 3D adaptation of the U-Net model to detect rib fractures (e.g. 3d convolutions). To benchmark FracNet, the authors also conducted 3D adaptations of an earlier Fully Convolutional Network (FCN) [9] and DeepLab v3+ [10]. The FCN only employs locally connected layers without dense layers, which reduces the number of parameters. DeepLab v3+ is another model for segmentation that builds on DeepLab v3 [11]. In general, DeepLab models address a criticial limitation of FCN, wherein images become smaller through convolutional and pooling layers and lose information. DeepLab v3+ uses Atrous Convolutions and Atrous Spatial Pyramid Atrous convolutions add the dilation rate parameter to the usual convolutional layer, which provides a wider field of view for any given computational cost (e.g. a 3x3 kernel with dilation rate=2 has the same field of view as a 5x5 kernel). Atrous Spatial Pyramid Pooling employs multiple parallel filters with different rates before convolution, which effectively allows images to be probed with complementary effective fields of view.

Compared to the 3D adaptations of FCN and DeepLab v3+, FracNet had superior performance with Dice 71.5%, IoU 57% and sensitivity 93%. A particular strength of this paper was that the authors evaluated real-world implementation of FracNet. They showed that although radiologists can detect rib fractures with higher sensitivity compared with FracNet alone, deploying FracNet (e.g. radiologist + FracNet) allowed faster rib fracture detection with higher sensitivity than radiologists achieved on their own. Bedside implementation should be the end goal of all AI-based medical imaging models; the highest performing models would have little value if they are not used by clinicians. Unfortunately, few AI-based models are used by clinicians at the bedside.[12] That the authors deployed their model and showed clinical utility was commendable.

Similar to Yao et al's paper, the authors only classified rib fractures binarily. There remains a critically missing need for clinicians to widely adopt an automated algorithm that can identify rib fractures: a model that can characterize rib fracture patterns, beyond detection. The aforementioned two tudies provide a framework we hope to replicate for our study, but our additional contributions that we specifically address will be outlined in the following sections.

3. Problem statement

We aim to identify and characterize rib fractures on chest CT scans using a deep learning model. The inputs will comprise dicom images (chest CT scans) from patients with known traumatic rib fractures. Ideally, the output will be a user-friendly, intuitive table which specified: 1) the total number of rib fractures; 2) the exact location of the rib

label code	description
-1	fracture of unknown type (ignored)
0	background
1	displaced fracture
2	non-displaced fracture
3	buckle fracture
4	segmental fracture

Table 1. Per-pixel label codes from the RibFrac challenge dataset.

fractures (e.g. left vs right, rib number [1-12], anatomic location [front-anterior, side-lateral, back-posterior]; 3) the degree of displacement [percent cortical contact between fracture segments]; 4) presence of flail chest [3 or more ribs broken in two or more locations]. The aforementioned characteristics have not been delineated by previous studies, yet are critical details that inform decision-making. In this work we make some progress toward these goals, namely a model the classifies fractures into sub-types, and a post-processing algorithm that calculates percent displacement.

4. Methods

We trained a 2D convolutional UNet model [13] on the RibFrac challenge dataset to predict an individual class label for each pixel in the sample images. The model is 2D in the sense that its input is a single 2D slice from a chest CT scan. The slices have a resolution of 512x512 pixels and each pixel is labeled with one of 6 classes. The class label descriptions can be seen in 1.

To prepare the data for training, we mapped all label ids to label codes and saved each slice of each CT scan in a separate file for efficient parallel data loading during training. We saved all "positive" slices with at least one non-background pixel in a folder called "pos" and all other "negative" slices in a folder called "neg." During training, mini-batches were constructed by taking 8 positive slices and 4 negative slices. This effectively under samples the empty slices and ensures that each batch has 8 slices containing fractures. Without this, we found the model tends to over-predict the background class.

We experimented with several different loss functions comprising different combinations of the cross-entropy loss and DICE scores, which will be further explored in the following sections.

4.1. Loss Functions

Cross-entropy (CE), a metric founded in in information theory, strives to measure the amount of change needed to transform an incorrect probability distribution to the ground-truth one of correct label classifications. It's functionally equivalent to "log loss" in classification; in both, the negative log of the softmax probability scores predicted for each label are joined in a convex mixture. This provides a highly effective loss metric that remains the state-of-the-art cost function for image classification models. The CE loss was computed as the mean cross-entropy loss over all pixels in the slice.

A key component of radiologists and clinicians reading CT scans to identify rib fractures is identifying where bones are in each scan, as well as if a fracture has formed in any bone. A visible boundary between two pieces of bone, in an area where there should not be such an abnormality, is an indicator of a fracture. In essence, encouraging our model to develop the ability to segment bones from surrounding tissue would intuitively improve model performance. To accommodate for this in our loss function, we decided to additionally incorporate 'DICE" loss. DICE is commonly used in CNNs for medical imaging, as well as for image segmentation and boundary detection tasks as a whole. It also helps mitigate "data imbalance," in which certain labels are significantly less common than others. [14] For DICE we computed binary-DICE (BD) and multi-DICE (MD) loss, where BD is the soft DICE score for background vs non-background pixels and MD is a simple average of the soft DICE scores for each non-background class. The -1 label was completely ignored and masked out in all cases. We also experimented with class re-weighting to compensate for the highly imbalanced number of pixels for each class in the training data.

In general, the loss for a sample slice with predicted class probabilities $\hat{y}$ and label y was computed as

$$L(\hat{y}, y) = w_{ce}\, CE(\hat{y}, y) - \log BD(\hat{y}, y) - \log MD(\hat{y}, y)$$

where CE is the cross-entropy loss, and BD, MD are the BD and MD losses respectively. w_{ce} is a hyper-parameter that was set to 10. The reason for including the BD term and the w_{ce} weight is because the MD loss was found to fall into a local minimum in which the model predicts all background labels, because a given class c only appears in a fraction of samples in each batch, on average.

4.2. Class Re-weighting

For class re-weighting, we tested the commonly-used inverse class frequency and also the inverse square-root of the class frequency. In the latter case, the weight of the CE loss given to class c is

$$w_c = \frac{1}{\sqrt{N_c + 1}}$$

where N_c is the number of pixels with label c in the training set. More generally w_c can be expressed as

$$w_c = \frac{1}{(N_c + 1)^{\beta}}$$

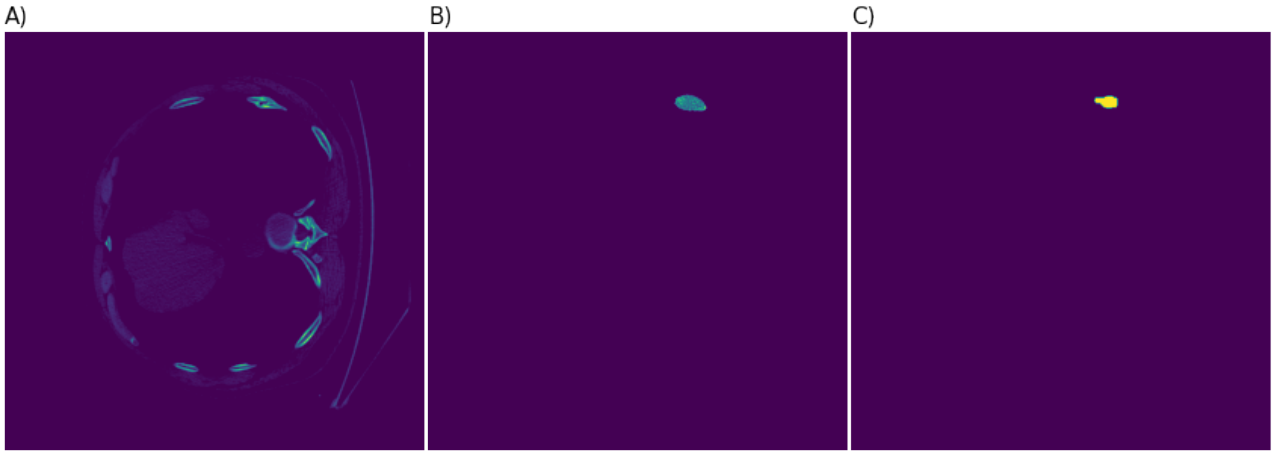

Figure 1. A qualitative example of the model prediction outputs. A) A single CT scan slice from the validation set. B) The 2D UNet model prediction. C) The ground truth label.

where $\beta \in [0, 1]$ is a hyper-parameter. In this work, we considered three values of β: 0, 0.5, and 1. $\beta = 0$ corresponds to no re-weighting and $\beta = 1$ corresponds to re-weighting by the inverse class frequency. The effect of setting $\beta = 0.5$ (square-root of the inverse class frequency) is a smoothing of the weight distribution as described in [15]. With $\beta = 1$, we found the loss curve became erratic as the loss changed drastically from one batch to the next and training took longer to converge. The weight of the -1 label, w_{-1}, was set to 0, and then the weights were normalized to sum to 1 for numerical stability.

4.3. Model Architecture

We now develop the model that will be evaluated with the 2 loss functions and 2 class re-weighting schemes. We decided to use a U-Net model, consisting of several convolutional layers that form a down-sampler and an up-sampler. The down-sampler is formed by a feed-forward, sequential set of convolutional layers of decreasing size and increasing filter counts. The up-sampler makes use of learnable up-sampling and max unpooling to recover the information encoded in smaller inputs in the middle of the model. [16] [17] Because U-Nets are used so ubiquitously in semantic segmentation tasks, and our model strives to predict fractures in certain regions of the image, we thought using one would be very effective. In addition, by increasing the number of convolution operations for smaller filters and down-sampled images, we'll preserve the latent representations of our model while also reducing the computational cost of our architecture. The model architecture and test-time workflow is shown in Figure 4.3. The UNet will take in a (1-channel) CT scan image as input, process it with a series of down-sampling blocks and up-sampling blocks in a feed-forward fashion, and then output an image in which pixels have the value codes presented in section 4. Each down-

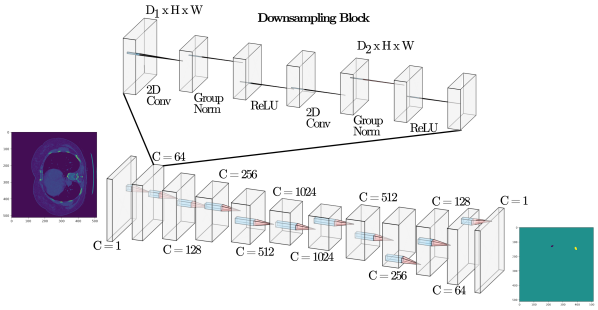

Figure 2. Model schematic for our UNet model. Core dimensions shown for each Block layer. Each block in the left half of the CNN is a down-sampling block ($D_2 < D_1$), the architecture of which is shown above, and a max pooling layer. Each block in the right half of the CNN (where layer sizes increase) are up-sampling blocks, which consist of a transposed 2D convolution and down-sampling block (except with $D_2 > D_1$).

sampling block consists of a 2D convolution step with an increasing number of filters (channels), group normalization, ReLU activation function, all repeated twice. In addition, between each down-sampling block, a max pool (2x2) layer is added. The original input is down-sampled until the block outputting a tensor with depth 1024. Then, up-sampling begins, where the output is slowly expanded back out into a tensor of depth 1 (the output) image. The only difference between the blocks used for up-sampling here and the down-sampling blocks are that an additional, 2D transposed convolution layer is used.

4.4. Percent Displacement Calculation

In addition to the dirth of a clinical-grade automated tool that can recognize the presence and type of rib fractures from chest CT scans, there are currently no tools that can characterize the percent displacement of a displaced rib

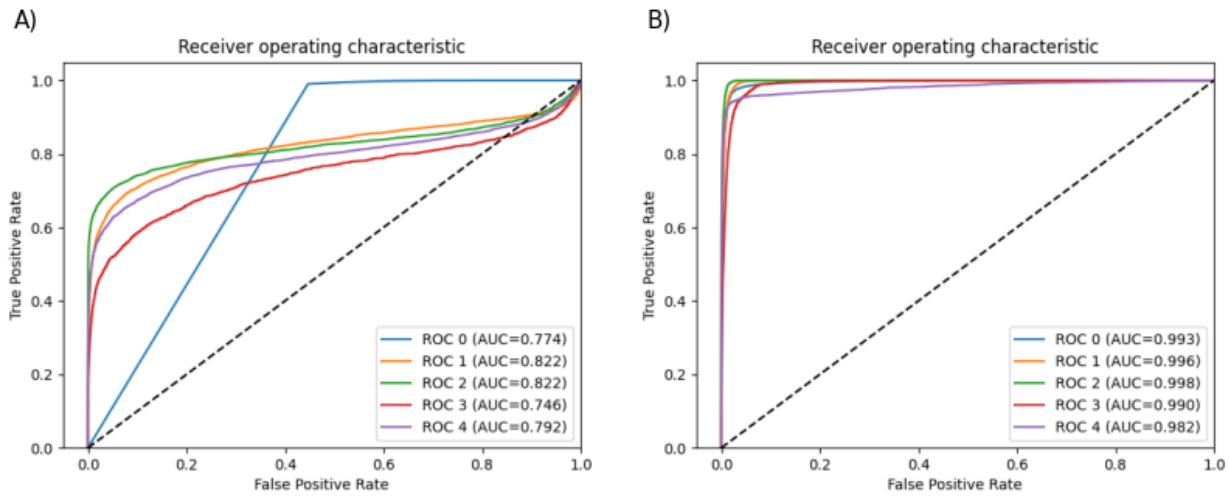

Figure 3. A) ROC curves computed for the CE+BD+MD model without class re-weighting. B) ROC curves for same model except with class re-weighting ($\beta = 0.5$). These ROC curves were computed based on a random sample of 500 slices from the validation set.

fracture. Displaced rib fractures, discussed in section 4, are a specific type of rib injury in which two fragments of a rib are offset by a certain percentage. A displacement of 0% corresponds to two fragments remaining exactly where they were in the rib before the fracture, but now with a break between them, and 100% corresponds to one fragment being completely out of line from the rest of the rib (and the other fragment). Identifying the precise percent displacement is extremely useful for clinicians and radiologists to provide accurate prognosis and treatment plans.

We developed the first automated tool to find the percent displacement (and existence) of a displaced rib fracture. After training the U-Net model on our dataset, we developed an *a priori* algorithm that is run on all of the CT scans identified to have a displaced fracture by the U-Net model. The algorithm takes in all scans that correspond to the same fracture, which were usually contiguous. It starts by preprocessing inputted chest CT scans. Specifically, all pixels that were not within a certain radius of pixels labeled to contain a displaced fracture were masked out (set to have a value equal to air/background, and thus completely ignored by the algorithm). It also "sharpens" the image. Although several, more complicated techniques to sharpen an image (convolving with a sharpening filter and additional non-linear methods) were used, none were more effective than again masking out all pixels with a value within 25% of background pixels. This provided a sharpened region of the CT scan that contained the displaced fracture. Then, a randomized Breadth First Search (BFS) algorithm was developed to find the interior and exterior regions of the bone fragments. At each iteration, a random point was chosen from all remaining pixels that had not been characterized. Since pixels on the interior of rib bone fragments had the same value as background pixels, this initial pixel was either characterized as a "background" pixel or an "edge" pixel (inspired by edge detection, as these pixels were on the exterior of each bone). Then, surrounding pixels were explored in a BFS fashion (also imposing boundary conditions) to probe both exterior and interior regions of rib fragments. Once a region was completely explored, the algorithm would remove all explored pixels from a list of those yet to be characterized, and repeat from the beginning. After all pixels were explored for each chest CT scan slice, the algorithm had obtained the exact positions of all of the pixels corresponding to each bone fragment across all CT slices. To obtain an accurate representation of the entire displaced fragment across multiple CT scans, the algorithm then merges these bone fragment pixel groupings across the different scans, producing a final image of the pixels, and thus position, cross-sectional area, and orientation, of each bone fragment. Once the positions and orientations of the two bones were produce, solving a simple linear regression problem and calculating the angular offset of the two bones (factoring in the width of each fragment) yielded a fracture displacement percentage.

5. Data

5.1. Overview

Our dataset is derived from two sources: 1) A publicly-available dataset of 5,000 annotated rib fractures from 660 chest CT scans (420 training, 80 validation, 160 evaluation).[18] 2) non-annotated chest CT scans from 2,000 patients with known traumatic rib fractures admitted to Stanford hospital.

5.2. RibFrac dataset

After signing the data use agreement, we downloaded the RibFrac dataset to our Sherlock group folder. Sherlock is a high-performance computing cluster that provides computing resources for Stanford researchers, including up to 300TB of file storage space. After reviewing Jin et al's manuscript (which used the RibFrac dataset), we succesffully trained their model ("RibFrac") on the downloaded dataset.

We installed a Jupyter Notebook within the group folder to facilitate real-time collaboration and evaluation using images stored in Sherlock, without having to download images locally.

5.3. RibSeg dataset

The RibSeg dataset [19] provides rib bone segmentation labels for 490 of the CT images from the RibFrac Challenge dataset. We intend to train our own version of the published RibSeg model and potentially use it as part of our rib fracture detection and annotation pipeline. Time permitting, we may attempt to train our own rib segmentation model and compare its performance with RibSeg.

5.4. Stanford dataset

The Stanford dataset was provided by the Research Informatics group, as dicom files stored in zipped TGZ files. We wrote a script to unzip all TGZ files, and found each unzipped file contained subfolders whose names (e.g. 1TKJSIDJFSDFKDMFDF) could be linked to a unique patient id using a key that is accessible only by the IRB-approved, clinical member of the team (JC). This team member created a dictionary linking subfolder names to anonymized patient ids and radiologist reports detailing the ground truth rib fracture injury patterns.

Dicom images have to be converted to NIFTIs to facilitate analysis. We have successfully converted a sample of dicom files to NIFTI format, and have written a script that parses through all subfolders to convert dicom files into NIFTIs, and yield NIFTI with file names that will facilitate tracing back to the aforementioned anonymized patient ids for evaluating performance.

Of note, due to time constraints, we were not able to externally validate our models on the Stanford dataset; this pre-procesesd dataset will be used in our follow-up study.

6. Results

For each of the four training schemes, we computed the binary-DICE and macro F1 scores. The binary DICE score is the soft DICE score for fracture vs. non-fracture and was computed by summing the non-background softmax probabilities to get a fracture probability for each pixel. The macro F1 score is a simple average of the F1 scores for each of the non-background classes. These results can be seen in table 2.

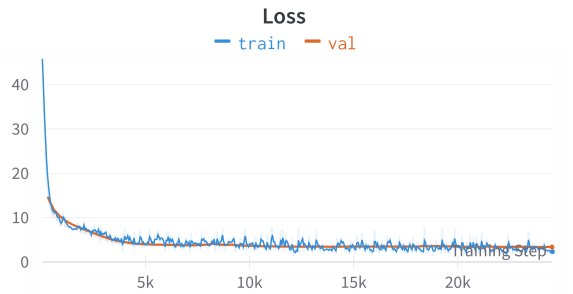

Figure 4. Training and validation loss curves over 5 epochs for the CE+BD+MC+RW-0.5 loss function.

We also experimented with re-weighting the cross-entropy loss for each class as described in section 4.2. Re-weighting significantly improved the F1 scores across classes, but seemed to cause the DICE scores to decrease, most likely due to an increase in the false positive rate since it gives far less weight to the background class.

Figure 6 shows the validation macro F1 score during training for identical loss functions except with and without re-weighting. This shows that re-weighting improves the macro F1 slightly, however figure 5 shows that it also decreases the binary DICE score due to the increase in false positives. We confirmed this qualitatively by examining the model predictions on the validation set and noticed that a much higher probability threshold was needed on re-weighted model to get accurate predictions. The effect of class re-weighting is especially pronounced in the ROC curves show in figure 3 where we see that it significantly improves sensitivity and specificity on all classes.

The loss curves for the CE+BD+MC+RW-0.5 model are shown in figure 4. It's clear the model is not over-fitting as the validation loss closely tracks the training loss, and we found this to be true of all loss functions tested. Furthermore, none of the models achieved extremely high performance metrics, even on the training set which suggests our model architecture may lack representational capacity. It is also interesting that even though the loss flattened out, the DICE and F1 scores continued to climb even at the end of our five epoch limit, as seen in figures 5, 6. Thus, it is conceivable that training a greater number of iterations, perhaps for 20 or 100 epochs may yield significantly better performance, but five epochs was all that our computational budget allowed. To verify our model had minimally sufficient capacity, we over-fit a single training batch and achieved near perfect accuracy, however it took approximately 200 training steps.

loss function	β (class re-weighting)	binary DICE	macro F1	epochs	learning rate
CE	0	0.0780	0.2396	5	1e-6
CE	0.5	0.0284	0.2248	5	1e-6
CE + MD	0.5	0.0251	0.2307	5	1e-6
CE + BD + MD	0	**0.8817**	0.2084	5	1e-6
CE + BD + MD	0.5	0.6572	0.2506	5	1e-6
CE + BD + MD	1	0.3900	**0.2649**	5	1e-6

Table 2. Shows the soft binary DICE (fracture vs no-fracture) score and macro F1 score achieved on the validation set for four different training schemes. The F1 score is an unweighted average over the four non-background classes. BD is the binary DICE and MD is the average DICE score for each non-background class (excluding the -1 class which is masked out of all performance metrics.)

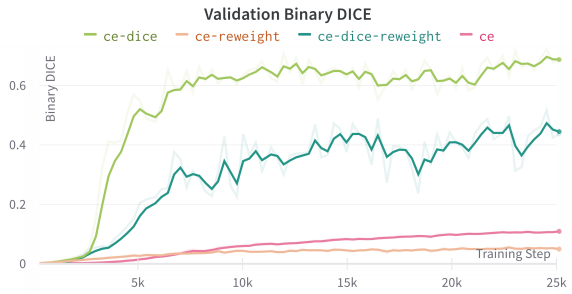

Figure 5. The binary (fracture vs no-fracture) DICE score on the validation set during training.

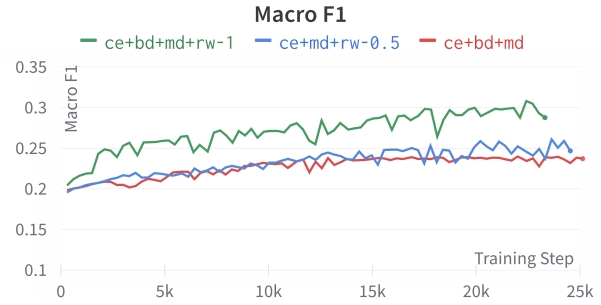

Figure 6. The macro F1 score across non-background classes DICE for the CE+DICE loss function with and without class re-weighting ($\beta = 0, 0.5, 1$).

6.1. Percent Displacement Calculation

To test the percent displacement algorithm, we applied it to one of the fractures that the U-Net successfully identified, shown in 7.

The algorithm was able to successfully characterize the fracture as a $> 100\%$ displaced fracture. The algorithm performed similarly well on a $< 100\%$ displacement example. The calculation was made robust to several different factors, including position in the overall image, overall rotation, bone width, and individual chest CT scan quality.

7. Conclusion

We have presented a 2D CNN model that is capable of accurately detecting rib fractures in chest CT scans, and a novel post-progressing algorithm which robustly calculates percent displacement based solely on the fracture position and the image. Our model has some difficulty classifying fractures into sub-types, but given our results from class re-weighting, we are confident that sub-types can be reliably classified from the prediction probabilities using a simple post-processing procedure. In particular, we suggest using blob detection on the output to first extract fracture candidates as blobs of pixels. For each candidate, one can compute the mean probability for each fracture sub-type over all pixels, and use pre-computed thresholds to classify it. Appropriate thresholds could be determined by ROC analysis.

This work involved a CNN model that was trained using several hyper-parameters, such as batch size, the degree of negative sampling, the β parameter for class re-weighting, and the weights of the terms in the loss function. We lacked sufficient time and computational budget to thoroughly explore this space, and we did not train all models to full convergence. This leaves an opportunity to train significantly more accurate models for rib fracture detection based on the same or similar techniques. A promising direction would be extending our model to 3D, where its input would be either multiple slices, or 3D sub-volumes from the CT scan. This approach would be similar to the FracNet model [5]. Another research direction would be exploring alternative model architectures such as the visual transformer [20], however, fracture detection is a highly local problem and may not benefit from global attention mechanisms.

Calculating percent-displacement as a post-process step in fracture detection has, to our knowledge, not been previously attempted. Our displacement calculator may significantly improve the clinical utility of CT fracture detection pipelines and we consider it a key contribution. Such automatic analysis of CT scans using a combination of deep learning and image processing algorithms promises to enhance radiology and make its practice more efficient, cost-effective, scalable, and accurate and thus benefit patients and healthcare professionals alike.

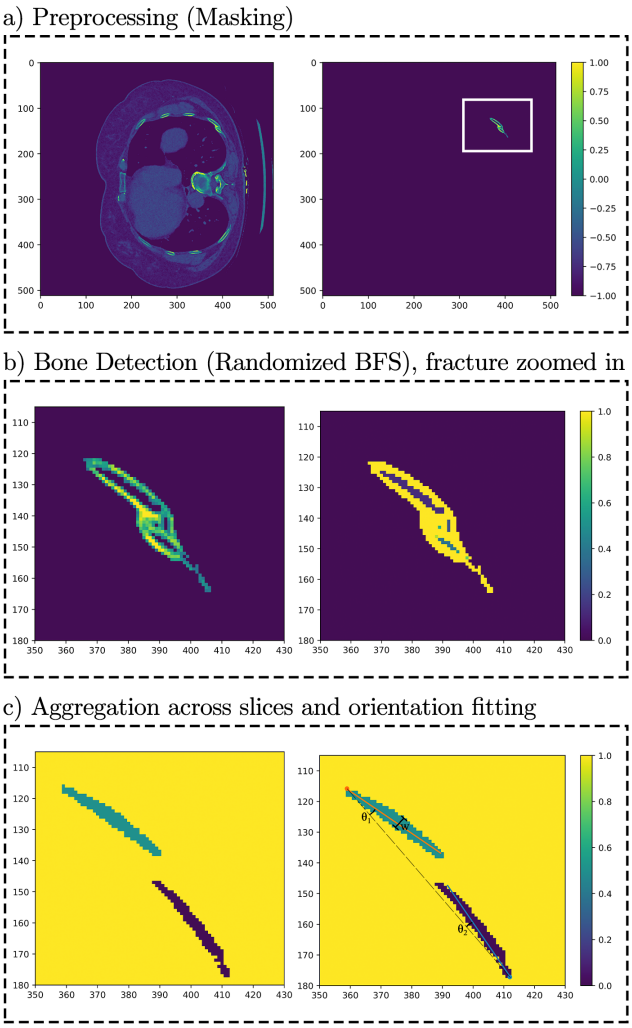

Figure 7. a) Pre-processing steps of the percent displacement algorithm, including masking out all pixels that weren't within a radius of the displaced pixels as identified by our U-Net model. b) Image sharpening and bone fragment pixel group detection; contiguous groups of pixels were determined to be groups of bones and very small groups of pixels were discarded. c) Aggregation of bone fragment pixel groups across chest CT scans and

8. Contributions & Acknowledgements

1. Jeff Choi
    - Used his medical background and research experience to provide guidance on the research direction, and the medical imaging data
    - Helped with data preprocessing, results analysis, and writing
2. Sathya Edamadaka
    - Developed the fracture percent displacement calculation algorithm
    - Helped with data exploration and pre-processing
    - Contributed to writing and produced figures for the final report
3. David Brown
    - Developed and trained the 2D UNet model
    - Wrote data preparation, model evaluation, and inference scripts
    - Ran multiple training experiments and collected data for analysis
    - Contributed to writing and produced figures for the final report.

References

[1] Jeff Choi, Aydin Kaghazchi, Katherine L. Dickerson, Lakshika Tennakoon, David A. Spain, and Joseph D. Forrester. Heterogeneity in managing rib fractures across non-trauma and level i, II, and III trauma centers. 1

[2] Jeff Choi, Giselle I. Gomez, Aydin Kaghazchi, John A. Borghi, David A. Spain, and Joseph D. Forrester. Surgical stabilization of rib fracture to mitigate pulmonary complication and mortality: A systematic review and bayesian meta-analysis. 232(2):211–219.e2. 1

[3] John G. Edwards, Peter Clarke, Fredric M. Pieracci, Mike Bemelman, Edward A. Black, Andrew Doben, Mario Gasparri, Ronald Gross, Wu Jun, William B. Long, Lawrence Lottenberg, Sarah Majercik, Silvana Marasco, John Mayberry, Babak Sarani, Stefan Schulz-Drost, Don Van Boerum, SarahAnn Whitbeck, Thomas White, and Chest Wall Injury Society collaborators. Taxonomy of multiple rib fractures: Results of the chest wall injury society international consensus survey. 88(2):e40–e45. 1

[4] Liding Yao, Xiaojun Guan, Xiaowei Song, Yanbin Tan, Chun Wang, Chaohui Jin, Ming Chen, Huogen Wang, and Minming Zhang. Rib fracture detection system based on deep learning. 11(1):23513. Number: 1 Publisher: Nature Publishing Group. 1

[5] Liang Jin, Jiancheng Yang, Kaiming Kuang, Bingbing Ni, Yiyi Gao, Yingli Sun, Pan Gao, Weiling Ma, Mingyu Tan, Hui Kang, Jiajun Chen, and Ming Li. Deep-learning-assisted detection and segmentation of rib fractures from CT scans: Development and validation of FracNet. 62. Publisher: Elsevier. 1, 7

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

[7] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017. 2

[8] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017. 2

[9] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2014. 2

[10] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018. 2

[11] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation, 2017. 2

[12] Jayson S. Marwaha and Joseph C. Kvedar. Crossing the chasm from model performance to clinical impact: The need to improve implementation and evaluation of ai. *npj Digital Medicine*, 5(1), 2022. 2

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 3

[14] Shruti Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE, October 2020. 3

[15] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples, 2019. 4

[16] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2016. 4

[17] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2014. 4

[18] RibFrac 2020 - grand challenge. 5

[19] Jiancheng Yang, Shixuan Gu, Donglai Wei, Hanspeter Pfister, and Bingbing Ni. RibSeg dataset and strong point cloud baselines for rib segmentation from CT scans. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, pages 611–621. Springer International Publishing, 2021. 6

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. 7