3D Rib Fracture Segmentation on Computed Tomography

Andrew Walker, Gaoxiang Luo*
Department of Computer Science and Engineering
University of Minnesota - Twin Cities
Minneapolis, MN 55455
{walk0655, luo00042}@umn.edu

Abstract

Rib fracture detection is a computer vision problem in medical imaging that the U-Net has achieved good performance in. This project explores a new design based on FracNet [15] as a baseline for fracture segmentation on 3D Computed Tomography (CT) volumes. The addition of attention mechanisms and the adaptation of more suitable model architectures (namely UNet++ [27]) improves on baseline performance, although we believe solely optimizing the segmentation network itself will not yield the best possible results. We show and discuss our experiment results, and outline plans for optimizing the entire pipeline for even better results as future work.

1 Introduction

Nearly every two out of three patients who experience chest trauma have rib fractures, and they are often associated with significant morbidity and mortality [24]. Usually, the number of rib fractures act as an important predictor for overall trauma severity and mortality for patients [20]. However, radiologists are often challenged to read hundreds of CT images and identify life-threatening injuries as quick as possible for every single patient [2], especially when hospitals are overloaded in unexpected circumstances (e.g., pandemic and disaster situations). In addition, a study shows that the missed rib fractures rate may be as high as 20.7% [4], which may lead to potential severe consequences for patients and clinicians. Having more radiologists involved in diagnoses may reduce the missed rib fractures, but it's not feasible in the area with radiologist shortage.

Therefore, a complementary approach utilizing deep neural networks (DNNs) has come into play and shown a certain level of promise [3] in clinical application, despite there being only a few deep learning algorithms for rib fracture detection from CT scans. To our best knowledge, several studies [26, 25, 15] have gained relatively good detection results, with 92.9% sensitivity [15], 91.5% specificity [25], and 91.1% accuracy with the assistance of radiologist [26] respectively. Notably, Jin et al. [15] were the first to formulate the detection task as a 3D segmentation task, with computation happening in 3D CT volumes directly instead of 2D CT slices as in other studies. In our paper, we have FracNet proposed by Jin et al. as our baseline, and aim to improve specifically the segmentation performance measured by Dice and IoU, especially for those elongated-shape fractures which the current FracNet failed to segment sufficiently [15]. Secondary considerations included training time and model memory footprint.

*These authors contributed equally to this work

Preprint. Under review.

2 Methods

2.1 Dataset

We used the RibFrac2020 contest dataset made open by Jin et al.[15], the first public thin-slice annotated rib fracture dataset, for this project. The dataset includes information for detection and classification, but we were focusing on just detection. The RibFrac dataset contains 500 annotated CT scans with 5,000 fractures total. The data is pre-split into training and validation sets, 420 and 80 respectively. The scans are 512x512x~300, with the distance between each scan being 1-1.25mm. Compared to commonly seen 1.5-5mm slices, this dataset's thin slices ensures that we are less likely to miss spatial context while applying 3D approaches. Also supporting this, it was clinically reported that thin-slice images could be helpful for the diagnosis of bone fractures [9]. Along with the CT scans, a rib fracture segmentation mask of the same size is provided. These segmentation masks were produced by multiple radiologists aided by a computer. The dataset consists of 63.8% men, 36.2% women, with an age range 21-94 with mean age 55.1 and standard deviation 11.82.

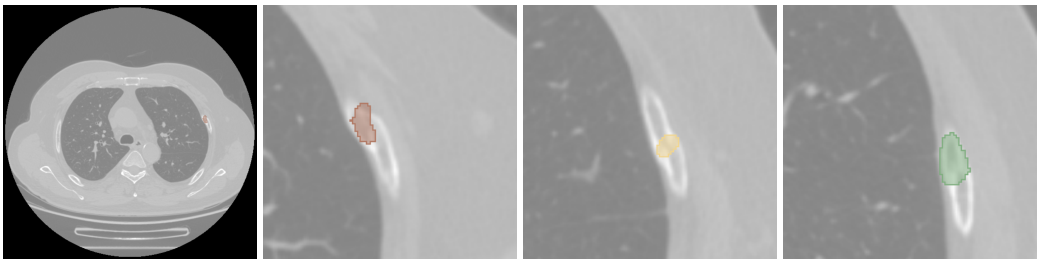

Figure 1: Visualization of an axial slice of a CT volume from the dataset and its corresponding slice of volumetric annotation with 3D slicer [6], an open-source software for CT scans and MRI images.

2.2 Network Architecture

Our baseline model is FracNet proposed by Jin et al.[15], the creators of the RibFrac dataset. The FracNet is an end-to-end segmentation system, consisting of preprocessing, the segmentation network, and post-processing. Their segmentation network is a standard 3D U-Net [28], and attempts to identify the rib fractures by performing voxel-level segmentation of the rib fractures on the CT volumes. The 3D context provided by a 3D convolutional network is helpful, as these thin CT slices have highly correlated information with the slices next to them. Additionally, U-Net is widely used in the medical computer vision community for segmentation tasks [23]. Notably, one of the highlights is that they formulated the problem as a segmentation task instead of a detection task, due to the fact that fractured ribs are usually long and thin, so using rectangular bounding boxes as a standard detection task may leave out much empty space within regions of interest.

The FracNet paper preprocesses the CT images by extracting the bone regions of the CT via morphological operations. This is a common practice done by clipping the CT measurement intensities to the "bone window" and min-max normalizing. This preprocessing allows for faster segmentation; we also used these same preprocessing transformations.

The baseline also conducts postprocessing of the FracNet raw output. It attempts to reduce false positives by removing predictions of small sizes and filtering the areas in which rib fractures could not exist (i.e. the spine). We did not touch the postprocessing strategy because our main focus was to improve segmentation performance, since overfitting was noted in the segmentation task in the original paper, especially for rib fractures of elongated shape.

In order to gain a better segmentation performance, we took multiple approaches. Our research was two-pronged: one direction was to modify the original FracNet architecture, and the other was to replace the network entirely. These two directions come together at the end as the final product, and both's results are provided and discussed. We

- Modified the baseline architecture with a number of features. These are referred to as baseline, prelu, se_at_end, se_throughout, and extra_layer.

    - baseline: This is the unchanged model as implemented in the original paper. It has 3 down layers followed by 3 up layers.
    - prelu: This is the baseline model with changed activation function by replacing ReLU's with PReLU's [11].
    - se_at_end: This is the baseline model with a squeeze-and-excitation attention module [13] inserted after the final Up layer. The SE module was adapted from its original 2D form to fit our 3D task.
    - se_throughout: This is the baseline model with squeeze-and-excitation modules after each Up layer.
    - extra_layer: This is the baseline model but it has one extra layer of depth.
- Replaced the underlying baseline U-Net with 3D versions of existing 2D U-like segmentation models, or implemented existing 3D segmentation models with some modifications. The models included FC-DenseNet [16] with "going-deeper" dense modules, nested U-Net (UNet++) [27] with short-connections as substitutes of long-connections, R2U-Net [1] with recurrent and residual mechanism, HighResNet [17] with many residual connections, Att-UNet [19] with attention gates, and CE-UNet [8] with dense atrous convolution (DAC) blocks and residual multi-kernel pooling (RMP) blocks.
- Reduced the depth of our best-performing model to explore potentially unnecessary computational waste.
- Implemented data augmentation to reduce overfitting by introducing Gaussian noise to the CT volumes, and flipping a small amount of CT volumes and volumetric annotations.

We implemented data augmentation because we noticed our models were exhibiting key signs of overfitting – towards the end of the 100 epochs in Figure 2 and Figure 3, training loss continues to fall while validation loss (and Dice coefficient) stays stagnant. We decided to implement data augmentation in the form of Gaussian noise and flipping to reduce overfitting, although as we will discuss, this did not seem to improve segmentation results.

Relatedly, the 25 n_epoch runs runs seemed to converge faster than the 100 n_epoch runs. This is an interesting result of the 1cycle policy's learning rate scheduling [22]. Notice that the 25 epoch and 100 epoch runs reach the same level of validation loss at the end of their runs, while the train loss of the 100 epoch run goes much lower. This discrepancy between the optimization of the decreasing training loss and steady validation loss is another sign of possible overfitting.

In the end, our final product combined the best of both research directions. We implemented squeeze and excitation modules throughout the entirety of UNet++. This attained a highest Dice Coefficient of 0.700964, while the vanilla UNet++ attained highest Dice Coefficient of 0.719293 and the SE throughout the normal FracNet achieved 0.6987. This is not an additive improvement on segmentation performance, and the implications of this reduction in performance even though each on their own improved performance will be discussed in the Discussion and Conclusion sections.

2.3 Evaluation Metrics

Since were intending to improve performance on the segmentation task, we used Dice coefficient and intersection over union (IoU) for evaluation, even though they're positively correlated. Notably, FracNet was originally reported to get 71.5% Dice coefficient on testing because it was trained on 720 CT scans, while we only had a subset of them (i.e., 420 CT scans which they made public) for training. Because of this, our FracNet implementation performance (68.5%) is slightly poorer than what was reported in their paper [15].

3 Experiments and Results

3.1 Model training

Since we didn't want to miss spatial context among CT slices, we applied all our networks on 3D CT volumes directly. It was fairly computationally expensive, especially since there weren't pre-trained models for 3D medical images yet. Thus, we used GPUs to enable more efficient training. For training, we used Google's Colab Pro service, the NVIDIA Tesla V100 GPUs server and the NVIDIA

Tesla K40 GPUs server from Minnesota Supercomputing Institute (MSI). We ran each model up to 100 epochs, which is almost the maximum time allowed in MSI for training (i.e., approximately 23 hours each).

We used the same training scheme used in the baseline paper. This was to randomly sample cubes of CT volume containing rib fractures and randomly sample cubes containing non-rib-fracture bone as training batches. During training, we used stochastic gradient descent (SGD) and a combo loss function composed of binary cross-entropy and soft dice loss with the weight ratio of 0.5:1 as the original paper did. We also used the same 1cycle learning rate policy [22] the original paper implemented, only changing the n_epoch parameter [15].

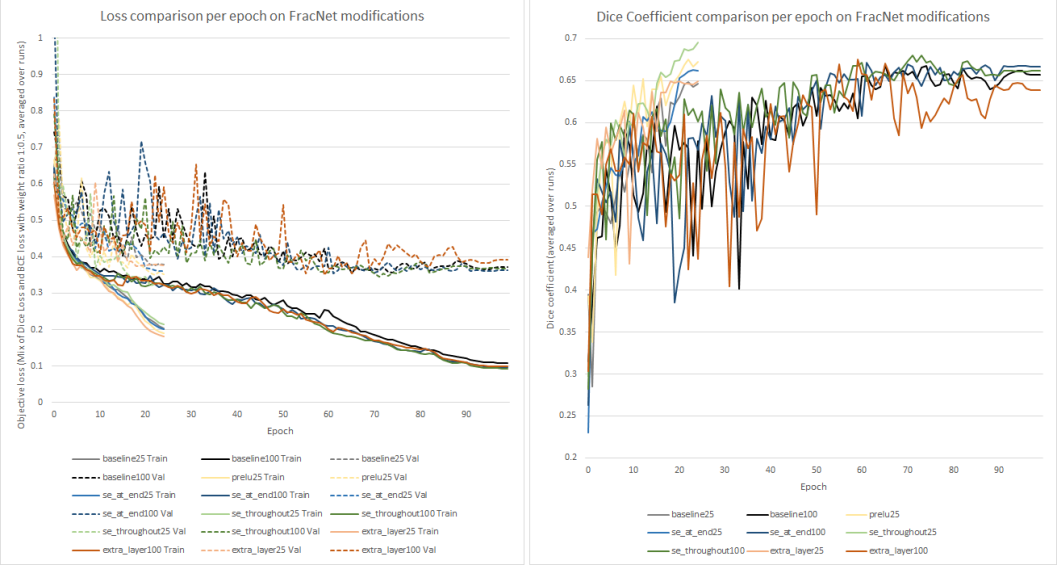

Figure 2: A comparison of metrics on modifications to the baseline FracNet model. The plotted metrics were averaged across multiple runs. Runs are split into two categories, indicated by the suffix of the series name. The 25 suffix indicates training with n_epoch=25, and 100 indicates training with n_epoch=100, in the fastai.Learner.fit_one_cycle method.

Table 1: A comparison of segmentation performance across the FracNet model modifications. Metrics are calculated on the validation set. We use the highest value across all training runs at any point during training. Modifications are split into two categories, indicated by the suffix of the column name. The 25 suffix indicates training with n_epoch=25, and 100 indicates training with n_epoch=100, in the fastai.Learner.fit_one_cycle method.

Modification	Dice (25 epochs)	Dice (100 epochs)
baseline	0.6832	0.6847
prelu	0.6751	None
se_at_end	0.675	**0.6903**
se_throughout	**0.6987**	0.6841
extra_layer	0.6488	0.6752

4 Discussion

We're fully amazed by the progression from early encoder-decoder structure brought out by Hinton in 2006 [12], to early deep learning in medical imaging [5], to the birth of U-Net [21]. The topological structure of U-Net is simple and elegant, and it's empirically known that utilizing U-like schemes guarantees not-bad performance, but we may naturally ask why this may work for 3D segmentation tasks in medical domain. Many studies show that downsampling in general can increase the robustness of the models (i.e., more invariant to translation and rotation, etc), and deeper layers in down sampling extract more abstract features due to larger receptive field while earlier layers extract

Table 2: A comparison of CT segmentation performance and model size among various neural networks. In order to have these models comparable to each other, they all have 16 feature maps after the first convolution, and reach 128 features maps after 3 more rounds of down-sampling except UNet++ that is a special case we'll discuss in later section.

Network	Dice	IoU	Params size (MB)	Speed (min/epoch)
FracNet	0.680220	0.525413	5.35	9.5
UNet++	**0.719293**	**0.566295**	419.01	13
FC-DenseNet	0.669421	0.510579	3.23	9.5
HighResNet	0.654656	0.495752	**3.07**	9.5
R2-UNet	0.610761	0.448741	72.27	12
Att-UNet	0.680837	0.521985	24.79	9.5
R2-Att-UNet	0.557510	0.393041	72.35	10
CE-UNet	0.656811	0.497747	17.36	9.5

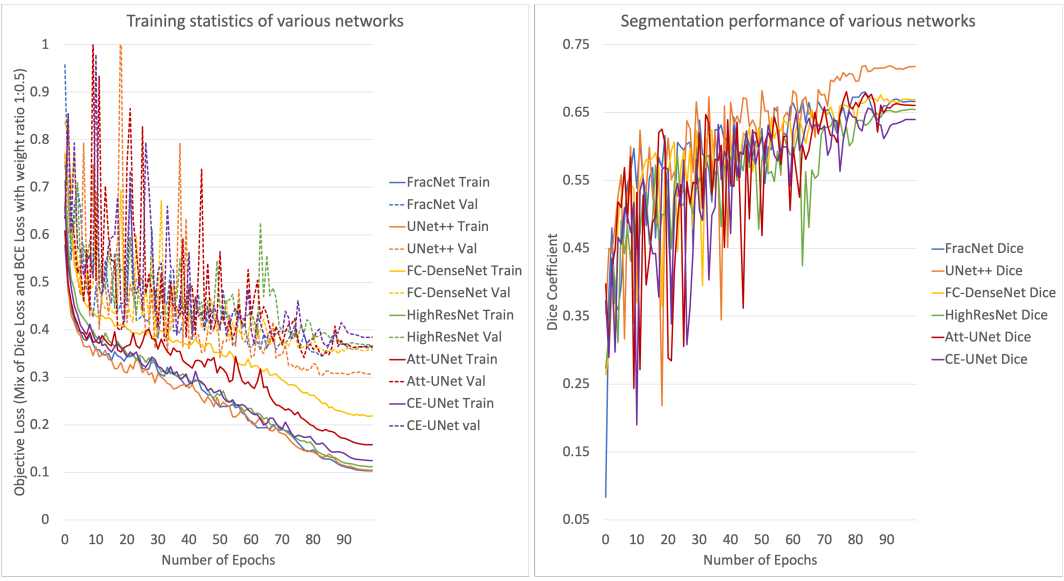

Figure 3: A comparison of 3D version of various segmentation networks in terms of objective loss and segmentation performance.

Table 3: A comparison of CT segmentation performance of UNet++ of different numbers of levels (i.e., depths, see Figure 4)

Network	Dice	IoU	Params size (MB)	Speed (min/epoch)
UNet++(L5)	**0.719293**	**0.566295**	419.01	13
UNet++(L4)	0.677537	0.516778	102.14	13
UNet++(L3)	0.688286	0.527761	23.65	12.5
UNet++(L2)	0.635777	0.470430	4.65	12.5

human-understandable features. Since both deeper and earlier layers can be important, we tried reusing the earlier features in deeper layers with dense modules. However, we found FC-DenseNet tended to converge first concerning training loss (see Figure 3), so it seems fusion of features at different depths confused the model, especially when the region of interest is small. A similar explanation can apply to HighResNet that we used here with residual blocks to enable the direct fusion of features on different scales.

If we think of both FC-DenseNet and HighResNet as attempting to enhance communication vertically (i.e., along the down sampling path), which failed to optimize the problem (see Table 2), then UNet++ can be looked at as enhancing communication horizontally (i.e., stronger commu-

Table 4: A comparison of CT segmentation performance of UNet++ and FracNet with and without data augmentation. Data augmentation steps were to introduce Gaussian noise to CT volumes and flip some 3D CT regions and corresponding volumetric annotations. Gaussian noise was added to the CT volumes after other preprocessing steps with mean 0 and variance 0.1. Flipping was done randomly to half of the data across the patient's sagittal plane. IoU was not recorded for FracNet.

Network	Dice	IoU
UNet++	**0.719293**	**0.566295**
UNet++ (Data Augmentation)	0.668518	0.511482
FracNet	**0.6832**	N/A
FracNet (Data Augmentation)	0.6779	N/A

Table 5: A comparison of CT segmentation performance of UNet++ and FracNet with different loss functions.

Network	Loss Function	Dice	IoU
FracNet	Combo Loss: BCELoss(w=0.5)Dice Loss(w=1)	0.680220	0.525413
UNet++	Combo Loss: BCELoss(w=0.5)Dice Loss(w=1)	**0.719293**	**0.566295**
UNet++	Combo Loss: Focal(w=0.5)Dice Loss(w=1)	0.689648	0.534226
UNet++	Dice Loss	0.708025	0.552059

nication to the corresponding neurons in the up-sampling path). The underlying intuition behind this enhancement is that we don't know which layers will turn out to be useful for this task but we can simply use all of them and let the model figure out which layers are important. Compared to the U-Net's long skip connections, which only concatenate features of the same level (i.e., depth), UNet++ concatenates the features of different levels together. We're convinced that different size of receptive fields on different levels have different sensitivity toward the segmentation target, and it's known that the small object and the edge information of large object are easily lost during repeated down-samplings and up-samplings. In our case, rib fractures, unlike organs, are fairly small objects, and this may explain why UNet++ performs better than FracNet. However, UNet++ has many more network parameters than FracNet, a standard U-Net. On one hand, once the training is done, UNet++ will theoretically only take approximately 30% more time to predict the rib fractures for a patient, which is around 130 seconds and acceptable. On the other hand, this may make UNet++ less accessible in application domains where computational resources are scarce.

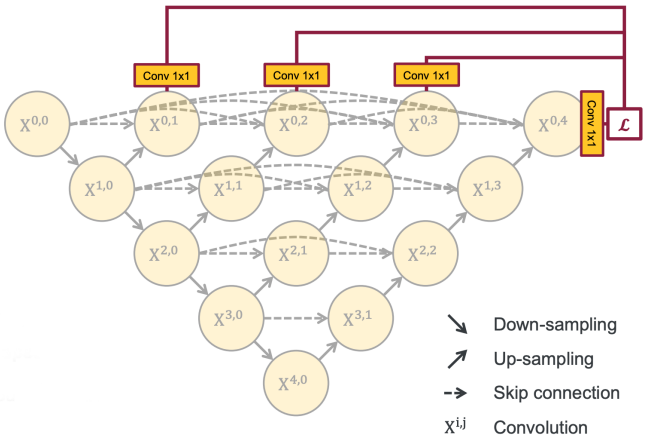

Figure 4: The network architecture of nested U-Net with emphasis of deep supervision, made by Zhou [27].

In fact, Zhou [27] used deep supervision to deal with the large-model issue, which we did not realize until the very end. Deep supervision here is to perform a convolution with kernel size of 1x1x1 for the output of each level. We know that back propagation happens only in training stage. If we finish training the model with a depth of 5 (see Figure 4), we're able to test the model by pruning

nodes $x^{4,0}, x^{3,1}, x^{2,2}, x^{1,3}, x^{0,4}$, since it will simply be the same model with a depth of 4 and the forward pass in testing won't update the internal weights of the model. Similarly, we can further test it on the depth of 3 and so on, to compare the performance of the model at different levels. If pruning turns out to have relatively the same performance as the non-pruning one, that implies the features learned in those levels that are pruned, aren't necessarily contributing to the segmentation of our target. Although in training, it's still a large model, this pruning allows for faster prediction, which is less resource-intensive for use in clinical practices. The model with deep supervision is still in training, but we'll continue exploring it during the winter break. For now, instead, if we say deep supervision is "first train then prune", we have the results of "first prune then train" in Table 3, which shows how performance varies with different model depths. Along the way, we also tried several loss functions that biases toward small object segmentation purposely. The result (see Table 5) showed a combo loss, composed of a distance measure (e.g., BCELoss) and overlap measure (Dice Loss), indeed worked better than the golden standard dice coefficient loss alone in segmentation tasks.

We would like to discuss two kinds of attention mechanisms that we implemented, which are adding squeeze and excitation blocks [13] and adding self-attention gating module [19]. Squeeze and excitation modules (SE) can adaptively weight the different model channels after feature concatenation of the Up U-Net network blocks [18]. It normalizes across channels and learns a re-weighting scheme. This helps give some channel features more weight than others in the final prediction. Attention gating modules can progressively suppress feature responses in irrelevant background regions, and aggregates information from multiple imaging scales, which ultimately will highlight salient features that are passed through the skip connections. Originally, we intended to rely on our attention modifications so that we didn't have to build an external localisation model for segmentation target area separation, but due to the relatively small performance gain it brought (see Table 1, 2), we would still work on a localisation model (e.g., a detector) to place before segmentation network, which we include in the future work.

We additionally encountered gradient explosion issue while training with R2-UNet and R2-Att-UNet (i.e., U-Nets include recurrent and residual operation) (see Table 2). These operations were designed to ensure better and stronger feature representation and thus help extract very low-level features, while maintaining the same number of parameters of U-Net. Alom's result on blood vessels (small object like fractures) segmentation was convincing [1], but our implementation for 3D input is questionable, which we will adjust throughout the break.

Separate from our architecture discussion, as mentioned in the experiments and results section, we noticed signs of overfitting during training. Our implemented data augmentation actually did not reduce this, it in fact caused the performance to go down even further. We are unsure of why this may be, but we hypothesize that it's more likely due to the Gaussian noise than the flipping. The fractures are already so small that the Gaussian noise's smoothing can really interfere with the features needed to identify where the fractures are.

As stated in the Network Architecture subsection, the components of our final prodcut (SE modules and UNet++) both outperform our baseline. However, the combined final product do not score higher than the UNet++ itself. It seems that segmentation improvements are hard to gauge and performance results are all very close. We take this to indicate that solely improving the segmentation network itself is not the best route to achieve the best improvements, and we should improve other parts of the pipeline as well to unlock as best a performance as possible.

5 Conclusion

Among all the plug-and-use modules/blocks into U-Net, attention mechanism seemed most promising and stable for tiny target like rib fractures. Although other modules enhanced the connection/communication within the model, the large background of CT volume could be a misleading factor to the model that negatively impacted the model performance. In addition, the plain nested U-Net outperformed FracNet in terms of Dice coefficient and IoU, but it increased the training time by approximately 30%. However, due to the pruning property with deep supervision of UNet++, the predicting/testing time on one CT scan didn't necessarily have to be as long as the training time after the model has been trained, so it still opened a door for potential application in clinical practice for a machine to find the fractures on a CT scan. Moreover, we realized simply modifying segmentation network wouldn't guarantee a significant improvement in performance, and many studies [14] have

shown that preprocessing is a very important procedure in medical imaging analysis. Therefore, in the following section we will discuss how we can do better to replace the current preprocessing mechanism used by Jin et al [15].

6 Future work

It seems that merely optimizing the segmentation network cannot improve the overall segmentation performance to a satisfactory scale, so throughout the break we plan to replace the clipping to bone window method in the preprocessing step with another neural network (e.g., feature pyramid network, RetinaNet) to perform detection first, and segment based on the results of the detector (see Figure 5). This is similar to Mask-RCNN proposed by He [10], a multitask model (i.e., first detection then segmentation). This aims to expose our segmentation network to an environment where it has higher chances to learn useful information (i.e., region of fractures), instead of clipping bone window that may have many cropping regions without fractures.

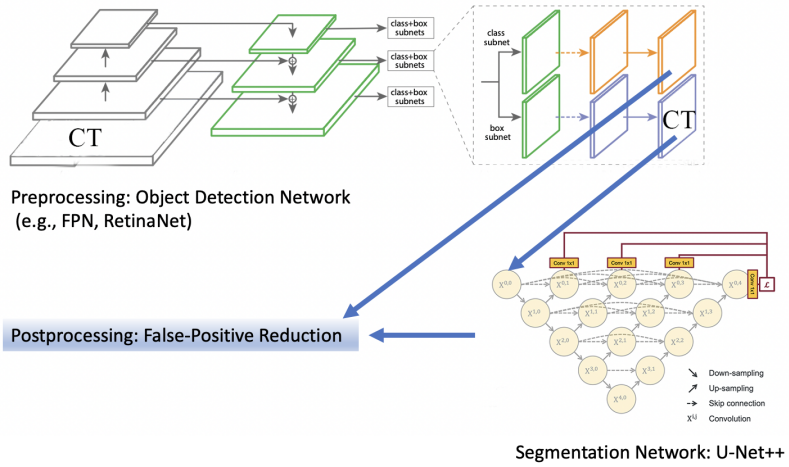

Figure 5: A blueprint of model pipeline for future work

Additionally, since having 2D CT slices as input may miss spatial context in volumetric space, and having 3D CT volumes as input suffers from computational intensity and memory limitations, we plan to discover a 2.5D approach. That is to combine several adjacent CT slices to be small CT volumes. In this way, we maybe able to utilize pre-trained 2D models on ImageNet [7] to reduce the training time and reduce overfitting on the small dataset while maintaining high performance.

7 Acknowledgements

This work discussed so far is part of a larger project that seeks to identify rib fractures and other pathologies in patients with just Chest X-Ray, CT scan, or both. This task with just CT scans for rib fracture may be used as part of the combined problem, or it may be used to help with data generation or transfer learning for the CXR-only problem. We hope that our overfitting problem we noticed will be mitigated with more data, which will be provided by Fairview Health Services.

References

[1] Md Zahangir Alom, Mahmudul Hasan, Chris Yakopcic, Tarek M. Taha, and Vijayan K. Asari. Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation, 2018.

[2] Nathan Banaste, Bérénice Caurier, Flavie Bratan, Jean-François Bergerot, Vivien Thomson, and Ingrid Millet. Whole-Body CT in Patients with Multiple Traumas: Factors Leading to Missed Injury. *Radiology*, 289(2):374–383, 2018.

[3] A Blum, R Gillet, A Urbaneja, and P Gondim Teixeira. Automatic detection of rib fractures: Are we there yet? *EBioMedicine*, 63:103158, 2021.

[4] S H Cho, Y M Sung, and M S Kim. Missed rib fractures on evaluation of initial chest CT for trauma patients: pattern analysis and diagnostic value of coronal multiplanar reconstruction images with multidetector row CT. *The British Journal of Radiology*, 85(1018):e845–e850, 2012.

[5] Dan C. Cireşan, Alessandro Giusti, Luca M. Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'12, page 2843–2851, Red Hook, NY, USA, 2012. Curran Associates Inc.

[6] James M. Curran. *Hotelling: Hotelling's T-squared test and variants*, 2013. R package version 1.0-2.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[8] Zaiwang Gu, Jun Cheng, Huazhu Fu, Kang Zhou, Huaying Hao, Yitian Zhao, Tianyang Zhang, Shenghua Gao, and Jiang Liu. Ce-net: Context encoder network for 2d medical image segmentation. *IEEE Transactions on Medical Imaging*, 38(10):2281–2292, Oct 2019.

[9] Leon Guchlerner, Julian Lukas Wichmann, Patricia Tischendorf, Moritz Albrecht, Thomas Josef Vogl, Sebastian Wutzler, Hanns Ackermann, Katrin Eichler, and Claudia Frellesen. Comparison of thick- and thin-slice images in thoracoabdominal trauma CT: a retrospective analysis. *uropean Journal of Trauma and Emergency Surgery*, 2018.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015.

[12] G.E. Hinton and R.R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science (New York, N.Y.)*, 313:504–7, 08 2006.

[13] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2019.

[14] Fabian Isensee, Jens Petersen, Andre Klein, David Zimmerer, Paul F. Jaeger, Simon Kohl, Jakob Wasserthal, Gregor Koehler, Tobias Norajitra, Sebastian Wirkert, and Klaus H. Maier-Hein. nnu-net: Self-adapting framework for u-net-based medical image segmentation, 2018.

[15] 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. *EBioMedicine*, 62:103106, 2020.

[16] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation. *arXiv*, 2016.

[17] Wenqi Li, Guotai Wang, Lucas Fidon, Sebastien Ourselin, M. Jorge Cardoso, and Tom Vercauteren. Information Processing in Medical Imaging, 25th International Conference, IPMI 2017, Boone, NC, USA, June 25-30, 2017, Proceedings. *arXiv*, 2017.

[18] Mehrdad Noori, Ali Bahri, and Karim Mohammadi. Attention and Multi-View Fusion. *2019 9th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 269–275, October 2019. arXiv: 2004.02009.

[19] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention u-net: Learning where to look for the pancreas, 2018.

[20] Jesse Peek, Yassine Ochen, Noelle Saillant, Rolf H H Groenwold, Loek P H Leenen, Tarsicio Uribe-Leitz, R Marijn Houwert, and Marilyn Heng. Traumatic rib fractures: a marker of severe injury. A nationwide study using the National Trauma Data Bank. *Trauma Surgery & Acute Care Open*, 5(1):e000441, 2020.

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv*, 2015.

[22] Leslie N. Smith. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay, 2018.

[23] Saeid Asgari Taghanaki, Kumar Abhishek, Joseph Paul Cohen, Julien Cohen-Adad, and Ghassan Hamarneh. Deep Semantic Segmentation of Natural and Medical Images: A Review. *arXiv*, 2019.

[24] Christopher J. Tignanelli, Alexander Rix, Lena M. Napolitano, Mark R. Hemmila, Sisi Ma, and Erich Kummerfeld. Association Between Adherence to Evidence-Based Practices for Treatment of Patients With Traumatic Rib Fractures and Mortality Rates Among US Trauma Centers. *JAMA Network Open*, 3(3):e201316, 2020.

[25] Thomas Weikert, Luca Andre Noordtzij, Jens Bremerich, Bram Stieltjes, Victor Parmar, Joshy Cyriac, Gregor Sommer, and Alexander Walter Sauter. Assessment of a Deep Learning Algorithm for the Detection of Rib Fractures on Whole-Body Trauma Computed Tomography. *Korean Journal of Radiology*, 21(7):891, 2020.

[26] Qing-Qing Zhou, Jiashuo Wang, Wen Tang, Zhang-Chun Hu, Zi-Yi Xia, Xue-Song Li, Rongguo Zhang, Xindao Yin, Bing Zhang, and Hong Zhang. Automatic Detection and Classification of Rib Fractures on Thoracic CT Using Convolutional Neural Network: Accuracy and Feasibility. *Korean Journal of Radiology*, 21(7):869, 2020.

[27] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 3–11. Springer, 2018.

[28] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: Learning dense volumetric segmentation from sparse annotation. *Lecture Notes in Computer Science*, page 424–432, 2016.