Chest X-ray Segmentation Ablation Study

Erick Platero (1812570; eeplater@cougarnet.uh.edu), Sneha Seenuvasavarathan (1996113; sseenuva@cougarnet.uh.edu), Dinesh Narlakanti (2083649; dnarlaka@cougarnet.uh.edu)

Department of Computer Science, University of Houston

1. Introduction

The medical field is often met with the challenge to identify new diseases, symptoms, and/or abnormalities in patients' bodies under different imaging modalities such as X-ray ultrasound, or near-infrared. These methods often require personnel with years of experience to generate virtual phantoms that mimic the geometry and size of the object of interest at a high resolution. Consequently, even though obtaining machinery that can extract images under different modalities becomes more accessible, having personnel that has the skills to create accurate virtual phantoms of the object of interest remains infeasible for medical facilities with scarce resources. This work seeks to mitigate this problem by training Deep Learning segmentation algorithms that can generate contrasted images of the object of interest. More specifically, this work will train five different Deep Learning segmentation algorithms: FCN, PSPNet, DeepLabV3+, SegFormer, and Unet. Each of these models take as input X-ray images of lungs with and without tuberculosis and outputs a contrasted image that segments the lungs (signal) from the background (noise). We chose to harness Deep Learning-based algorithms because these algorithms have surpassed human-level ability on image detection, localization, and segmentation tasks. Further, we train five different algorithms to analyze the different performance of each model under low-volume constraints of data. This is a significant factor as Deep Learning models require extensive amounts of data to be trained efficiently. This is in sharp contrast to the low-volume data found in the medical realm. As such, it is crucial to analyze the performance of different models to be able to make a decision as to what kind of deep learning model we want to deploy to segment our objects of interest. All code for this project can be found on the following GitHub repository: `https://github.com/eplatero97/LungSegmentationPerf`.

2. Dataset

To perform our experiments, we evaluate all our models on the Chest X-ray dataset that contains sources from the Shenzhen and the Montgomery dataset [1]. This dataset contains x-ray images of lungs and their segmented masks (see fig. 1. We chose this dataset because it's relatively small and is quite an accurate test case for our problem statement. This dataset was randomly partitioned into three sets: train (563), validation (71), and test (70).

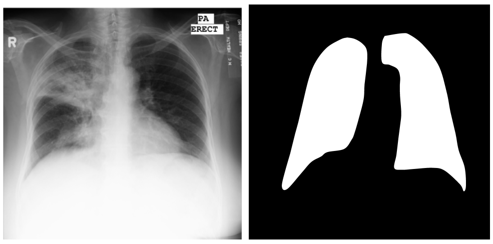
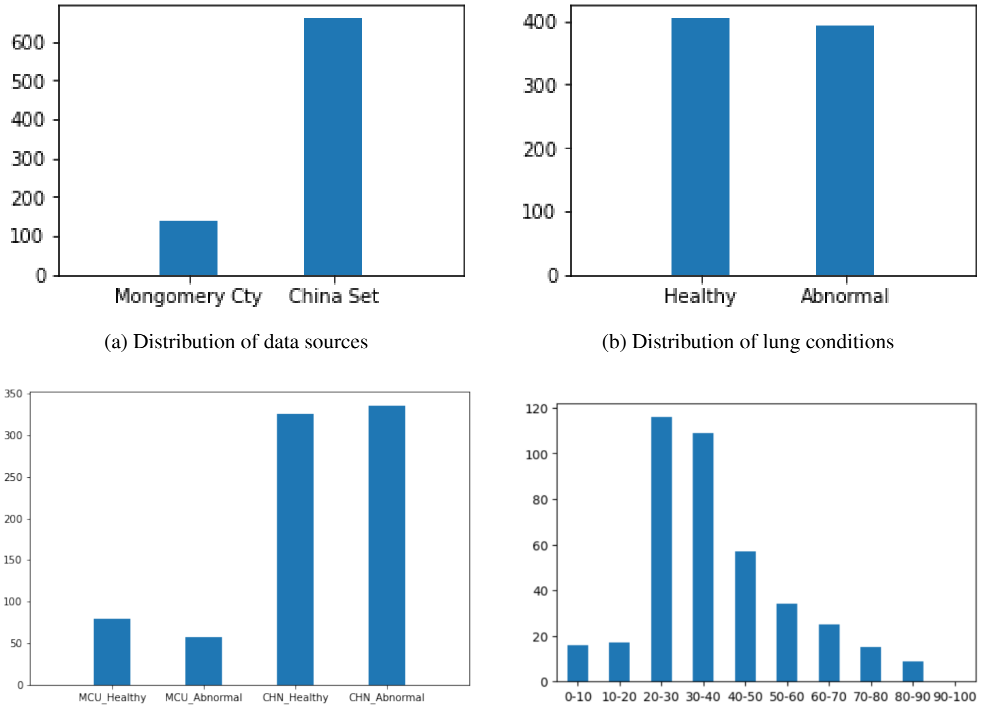


Fig. 1: left: lung x-ray; right: mask

2.1. *Exploratory Analysis*

Before diving into the segmentation task, we perform an exploratory analysis to understand the distribution of the data. The plots in figure 2 shown below represent the distribution of the data sources and the categories within the dataset. We observe an imbalance in the data contribution from two different data sources. Around 150 data entries are from the Mongomery County and around 650 data entries from the China set. Even though the contribution is imbalanced from the data sources, the distribution of healthy and abnormal lung-conditioned patients is balanced, which is an important factor for training. We used Data Binning to analyze the age of the patients and their abnormal lung conditions and came to the conclusion that the 20–40 year age group had the highest illness burden.



(a) Distribution of data sources

(b) Distribution of lung conditions

(c) Lung condition distribution across data sources

(d) Effect of age on presence of TB

Fig. 2: Exploratory Analysis

3. Models

The models that we are planning to use for the project are PSPnet [2], UNet [3], SegFormer [4], FCN [5], and DeeplabV3+ [6]. These different sizes contain different components (transformers, attention mechanism, convolutions) and vary widely in terms of number of parameters. As such, evaluating this set of models will allow us to get an idea as to what set of models perform test on our medical imaging dataset. In the following section, we describe the architecture and workings of the segmentation models.

3.1. UNet

U-Net is a U-Shaped symmetric encoder-decoder architecture that is able to localize areas of interest by performing classification on every pixel. The network consists of 4 encoders that extract features to learn abstract representations of the input, a connecting bridge that facilitates the transfer of information to the decoder, and 4 decoders to generate a semantic segmentation mask. Finally, a sigmoid-activated 1x1 convolution is performed on the decoder's output to obtain the segmentation mask that is classified pixel-wise.

3.2. SegFormer

This network combines the Transformer architecture with self attentive encoders that are hierarchical and multi-layer perception decoders(MLP). SegFormer has two main advantages (i) The network does not suffer from performance issues when the resolution of the training and testing datasets are different. This is attributed to the absence of positional encoding which mitigates the performance costs due to positional code interpolation. (ii) The MLP decoder employed in the network combines information from all the layers to render insightful representations. This network is known for its good performance despite its simple and light-weight nature.

3.3. DeepLabV3+

The DeepLabV3 model uses its backbone network to extract features from an image. The size of the feature map is managed by dialated convolution employed in the backbone network. Additionally, Atrous Spatial Pyramid Pooling (ASPP) [6] is applied to segment objects of different scales with better accuracy. The DeepLabV3+ model extends the DeepLabsV3 model by adding a decoder to the architecture to improve the segmentation results along the edges of an object.

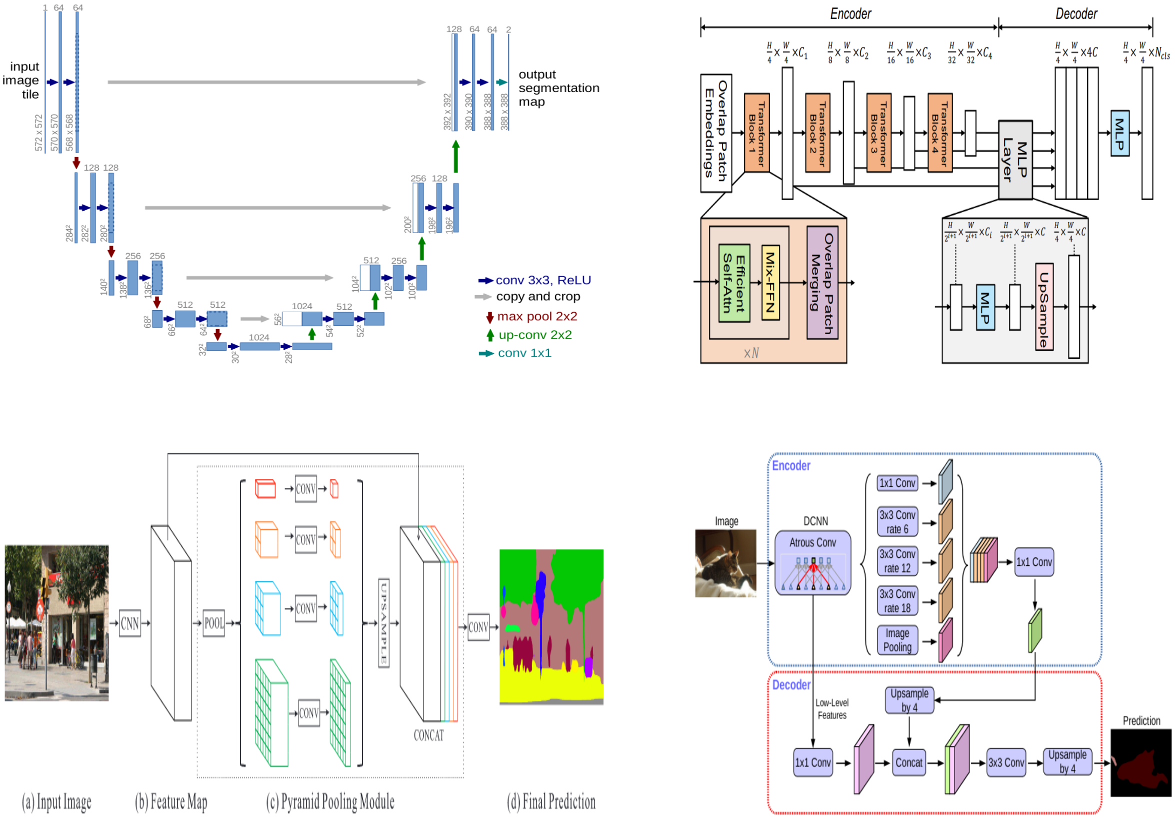


Fig. 3: top-left: UNet; top-right: SegFormer; bottom-left: DeepLabV3+; bottom-right: PSPNet

3.4. *PSPNet*

The architecture of the Pyramid Scene Parsing Network consists of two CNNs and a pyramid-pooling module. When the inupt image is fed into the model, feature extraction is performed by a CNN. This is followed by a pyramid parsing module that obtains the representations of all the composing regions. The final feature representation is formed by upsampling and concatenating the previous layer's output. The prediction is obtained by passing the final representation to a CNN.

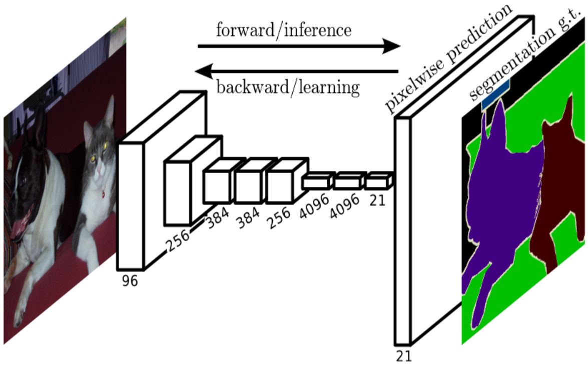
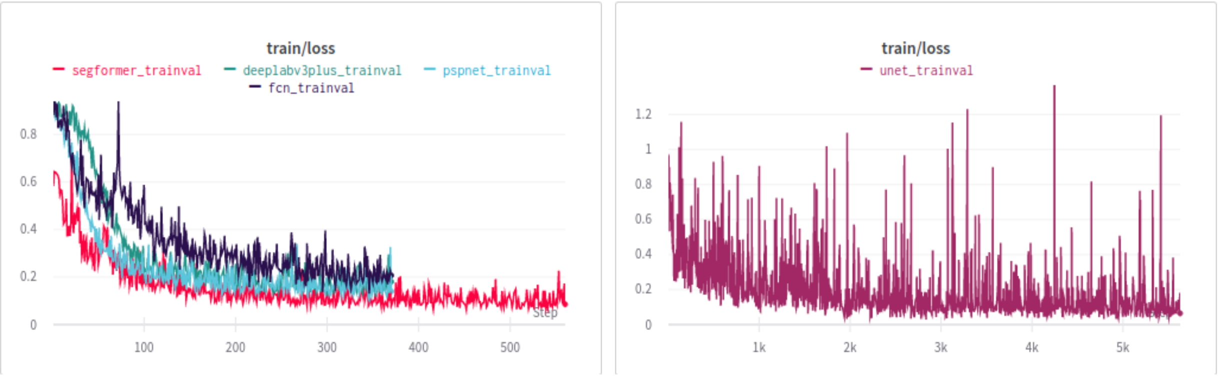


Fig. 4: FCN

3.5. *FCN*

Fully Connected Network(FCN) are made of locally connected layers such as convolution, pooling and upsampling. The absence of dense layer in this network makes prediction faster due to the reduced number of model



(a) FCN, PSPNet, DLV3+, and SegFormer

(b) Unet

Fig. 5: Train Loss

parameters. The architecture of this model combines semantic and appearance information from the deep and shallow layers to generate accurate segmentation results.

4. Ablation Study

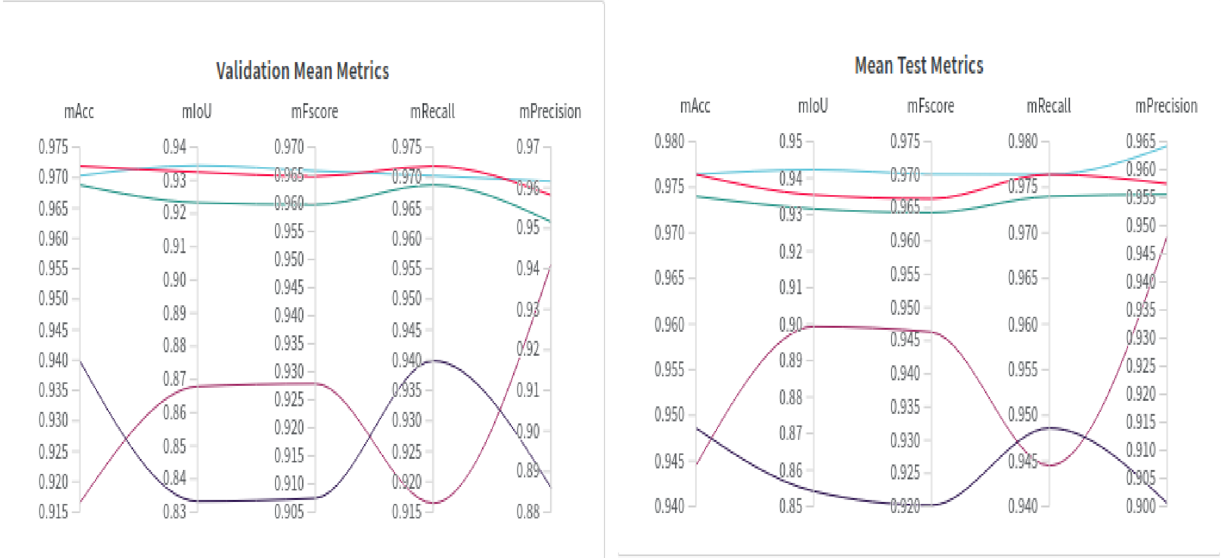
The ablation study followed a specific workflow: train each model for 10 epochs on the training dataset, evaluate each model after every epoch on the validation dataset, and test each model once after the 10th epoch on the testing dataset. We selected to train for 10 epochs to avoid the model from overfitting our training set and selected to test the model only once on the latest state of the weights to avoid the model being able to train on some weights that optimize the performance on the testing dataset.

4.1. Training

For each model, we use Stochastic Gradient Descent with a learning rate of 0.01, momentum of 0.9, and weight decay of 0.0005. Further, for the training pipeline, we pre-process the data by resizing the image to (2048, 1024), performing a random crop and random flip, distort the image, and finally normalize the image. For batch size, FCN, PSPNet, and DeepLabV3+ were trained with batch size of 15 while SegFormer was trained with batch size of 10, and UNet was trained with batch size of 1. The different batch sizes were due to memory constraints on the NCasT4_v3-series hardware provided by Azure services (https://learn.microsoft.com/en-us/azure/virtual-machines/nct4-v3-series). Due to this, the number of steps for UNet far surpasses the number of the steps in the rest of the models as seen in fig. 5. It is important to note that setting a batch size of one is expected to degrade overall performance and thus, this is a limitation of our work.

4.2. Validation and Test

For validation and testing, we recorded five different metrics to judge model performance: accuracy, precision, recall, fscore, and intersection over union. For both workflows, we demonstrate model performance on the state of weights after the 10th epoch training, which is shown in section 4.2 and section 4.2. The validation performance of each of the metrics on all 10 evaluations is shown in fig. 7. To be able to better visualize data performance per metric, fig. 6 shows parallel coordinates with the different metric performances. From these plots, we see that test performance followed the same general patterns as the validation performance. On the test evaluation, PSPnet and SegFormer both produced the highest levels of accuracy, both scoring 0.9764. The highest precision score in the test results belongs to PSPnet. With ratings of 0.9764, PSPnet and SegFromer have the highest recall rates. PSPnet is the top model with the highest fscore and also has the highest score for the IoU metric(0.9423). Since recall and precision are inversely related, it is impossible to have high recall and precision. At thresholds, recall is strong in proportion to low accuracy, and at very high recall, precision starts to decline. All the models considered exhibit the same tendencies. Additionally, PSPnet and Segformer have the highest Precision and Recall scores, respectively, when segmenting using the validation set. FCN produces the least precision score, whereas the Unet model produces the least recall score. As precision and recall are averaged (harmonic mean) to create the Fscore. Evidently, the center ground is where things stand. SegFormer has the highest Fscore of all the models taken into consideration, whereas FCN has the lowest Fscore. The most intuitive performance metric is accuracy. Due to the symmetry of our data set, we can determine which model is more accurate by looking at it. SegFormer has the highest accuracy out of the validation set, followed by PSPnet. Unet has the least accurate model. The



(a) Validation Performance

(b) Test Performance

Fig. 6: Performance Metrics. Red: SegFormer, Blue: DLV3+, Green: PSPNet, Black: FCN, Purple: UNet

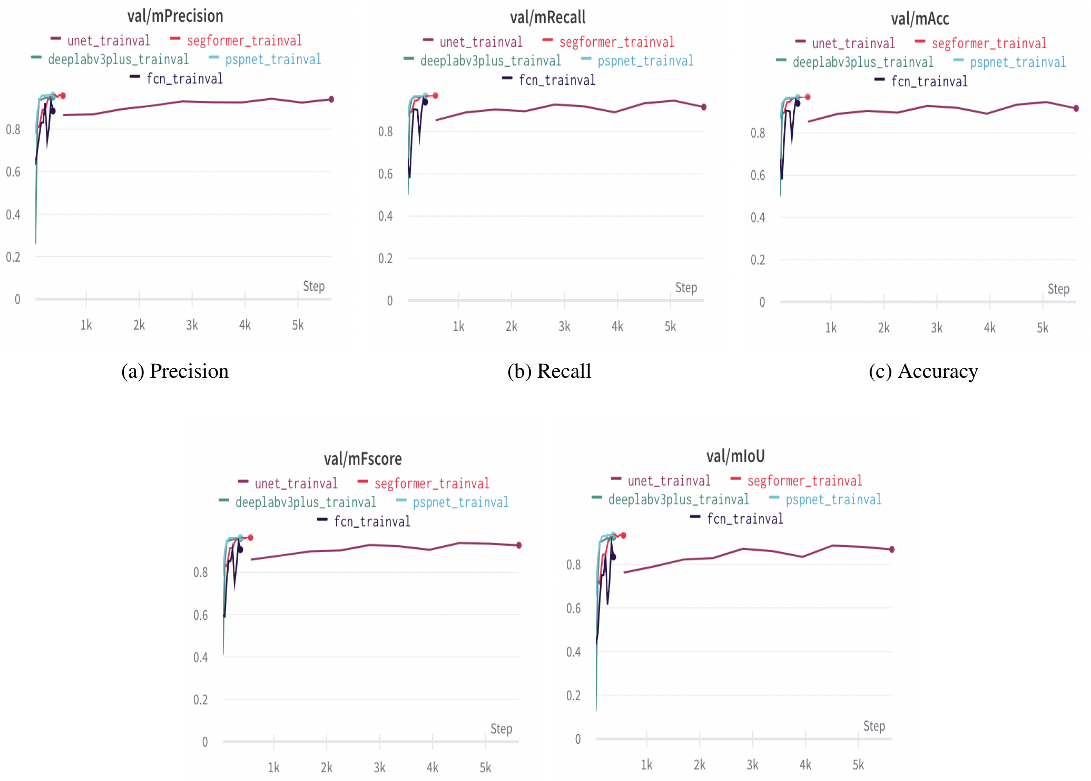
high performance of PSPNet may be because it considers both the local and global contexts of the image when making local level predictions. As for the worst performance of UNet and FCN, UNet's performance is likely due because it was by far the model with the most weights (loading model took more than 16GB, hence the batch size of one due to memory constraints). Such a big model is expected to require massive amounts of data to be trained properly. Further, given that it was trained on a batch size of one, this likely also degragates the overall performance of the model. As for FCN, the simple architecture might prevent it from learning complex patterns with low-volume data.

Validation Mean Metrics					
Model	Accuracy	Precision	Recall	Fscore	IoU
PSPnet	0.9703	**0.9616**	0.9703	0.9658	**0.9345**
SegFormer	**0.9719**	0.9582	**0.9719**	**0.9719**	0.9325
Deeplab	0.9688	0.9516	0.9688	0.9598	0.9233
Unet	0.9165	0.9411	0.9165	0.9279	0.8679
FCN	0.9399	0.886	0.9399	0.9075	0.8333

______________	Test Mean Metrics				
Model	Accuracy	Precision	Recall	Fscore	IoU
PSPnet	**0.9764**	**0.9642**	**0.9764**	**0.9701**	**0.9423**
SegFormer	**0.9764**	0.9575	**0.9764**	0.9664	0.9354
Deeplab	0.9742	0.9556	0.9742	0.9643	0.9316
Unet	0.9445	0.9482	0.9445	0.9463	0.8993
FCN	0.9486	0.9006	0.9486	0.9202	0.8512

5. Conclusion

From our analysis, we determine that PSPNet is the best segmentation algorithm to leverage segmenting lugs under low-volume data constraints. It may be that its Pyramid Scene Parsing operations are best suited to generalize low-volume data (with SegFormer and DeepLabV3+ being suitable substitutions). This analysis demonstrates that under low-volume constraints, choosing what architecture with what operations is a crucial questions as it can lead to very different performances. Further, this work also highlights that even if we automate the generation of a contrasted image via a model, the model will have inherit errors on its own. As such, it is never to be expected that these models will always be suitable replacements to generating human-made contrasts via phantoms. If a medical entity is to deploy a Deep Learning solution, then the benefits of the model must be weighted with its cons of not having any ability to explain its results and its expected errors.



(a) Precision

(b) Recall

(c) Accuracy

(d) FScore

(e) IoU

Fig. 7: Validation mean metrics for the five models

References

1. A. S. W. Y. L. P. T. G. Jaeger S, Candemir S, “Two public chest x-ray datasets for computer-aided screening of pulmonary diseases,” Quant Imaging Med Surg (2014).
2. H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *CVPR,* (2017).
3. O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention,* (Springer, 2015), pp. 234–241.
4. E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, “Segformer: Simple and efficient design for semantic segmentation with transformers,” arXiv preprint arXiv:2105.15203 (2021).
5. E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” IEEE transactions on pattern analysis machine intelligence **39**, 640–651 (2017).
6. L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *ECCV,* (2018).