Deep Learning for Pneumothorax Segmentation

Théo VIEL and Mathieu ORHAN

MVA, Ecole des Ponts ParisTech
{theo.viel, mathieu.orhan}@eleves.enpc.fr

Abstract. Pneumothorax is a lung disease causing breathing difficulties and pain, that can be fatal if not correctly treated. We study a supervised deep learning approach to detect pneumothorax on chest X-ray scans. We jointly learn segmentation and classification, to detect whether the patient has the disease and where it is located. We first start by using an usual baseline that has proven to be efficient, and then study several improvements and report their performances. We evaluate our experiments on the SIIM-ACR Pneumothorax Segmentation dataset with both the accuracy and the dice coefficient. The code is publicly available at `https://github.com/TheoViel/pneumothorax_segmentation`.

Keywords: Pneumothorax · Deep Learning · Segmentation · Classification.

1 Introduction

1.1 Pneumothorax

A pneumothorax corresponds to an abnormal presence of air between the lungs and the chest wall. It provokes difficulties to breath and pain in one side of the chest, and results in a poor oxygenation. It is worsening over time and, unless correctly treated, fatal. An accurate and fast detection is thus extremely important.

It is sometimes hard to diagnose it for an human expert, in particular for the smaller ones. Several modalities can be used for that purpose, such as CT scans, chest X-rays, or ultrasound. On X-rays, air corresponds to the darkest parts. To diagnose, one must try to delimitate lungs, which also contains air, with an hypothetical surrounding pneumothorax, usually at the top of the lungs. The separation, if one, is usually difficult to identify. However, the most advanced cases – that are medical emergencies – are quite easy to diagnose, as the lungs are very asymmetric and shifted. Finally, some other entities can be confused with a pneumothorax, mainly pulmonary bullae and hemothorax.

In this project, we propose to use a deep learning approach to detect pneumothorax on chest X-rays, using the SIIM-ACR pneumothorax Segmentation dataset. Chest X-rays can be treated as grey-scale images and processed with convolutional neural networks. At its core, our architecture will have the shape of a U-Net [16], a solid baseline for segmentation purposes. Pneumothorax diagnosis clearly requires fine-grained models that are able to identify the presence

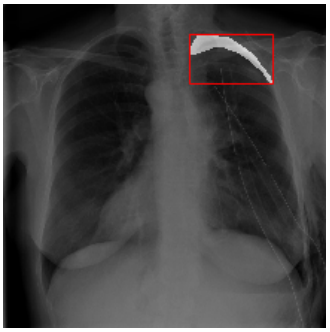


Fig. 1: Example of a positive sample. The pneumothorax area is highlighted.

of a pneumothorax, and if one, segment the evidence of it. We will introduce and experiment with different processing, designing and training strategies that can improve the performance of the detector.

1.2 Problem Definition

The high-level task is to use an image I to predict the probability that a patient is positive, and provide an evidence of the pneumothorax by segmenting out the pixels. The task is formally to jointly learn a function $\mathcal{F}_\theta$:

$$\mathcal{F}_\theta : I \rightarrow \{(p_i)_{i \in N}, p\}$$

where p is the probability of the patient being positive given I, N the number of pixels, and p_i the probability that the i-th pixel is a pneumothorax given I. This function is learned through back-propagation on the parameters θ, in a typical deep learning setting.

1.3 Evaluation

We report a metric d_{cls} and d_{seg} to evaluate the two tasks of classification and segmentation. Let's note $\hat{y} = \mathbf{1}_{p>0.5}$ and $\forall i \in 1...N, \hat{y}_i = \mathbf{1}_{p_i>0.5}$ the binary predictions. We denote by y and y_i the corresponding groundtruth. For the segmentation task, our goal will be to optimize the Dice coefficient, which is defined as:

$$d_{seg}((\hat{y}_i)_{i \in 1...N}, (y_i)_{i \in 1...N}) = \frac{2 \times \sum_i \hat{y}_i y_i}{\sum_i y_i + \sum_i \hat{y}_i}$$

It evaluates the agreement between positive pixels. If Y_{truth} is empty, we the dice is set to 1 if and only if the predicted mask is empty as well. Otherwise, it is set to 0. The goal is to heavily penalize false positives.
For the classification task, we use the accuracy with a threshold at 0.5:

$$d_{cls}(\hat{y}, y) = \mathbf{1}_{\hat{y}=y}$$

Qualitative results will also be provided in appendix, with a visualization of the results of the studied methods on two samples.

2 Related Work

The first method to automatically detect pneumothorax on chest X-ray scans is [7]. They first try to detect and model the rib cage, an important anatomical reference. They detect edges using a Sobel operator and fit a function with prior shape to model the ribs. This enables the to isolate and clean candidate regions for pneumothorax. The more subtle pneumothorax pattern is then detected using a gradient histogram analysis and inverse Hough transform.

The problem can be approached with atlases and deformable registration. [3] tries to find the closest training image with its masks in the training set, using a partial Radon transform and Bhattacharyya shape similarity measure. Then, SIFT-flow is used to obtain an initial patient-specific mask. This mask is finally post-processed using a graph-cut.

The problem can also be treated as supervised learning. Texture analysis (such a Local Binary Patterns or Maximum Response filters) are used to extract features used by classifiers such as Support Vector Machines [4] or K-Nearest Neighbors [8]. A recent method [15] uses fully convolutional neural network and outperform the previous works.

The trend is toward less anatomical priors and handcrafted features, larger datasets that can be used in a supervised learning setting. Our work is close to this last method [15], which makes use of several architectural choices to improve results, such as spatial and channel squeeze and excitation module, a multi-scale module, and a spatial weighted cross-entropy loss function. Our architectural proposals are complementary to this work.

3 Methodology

3.1 Data

The dataset used is the SIIM-ACR Pneumothorax Segmentation dataset, made available by the Society for Imaging Informatics in Medicine (SIIM)[1]. The data consists of 12000 chest radiographic images of size 1024×1024. Data is labelled with $y = 1$ if the patient has pneumothorax and $y = 0$ otherwise. On regions where the pneumothorax is located, the pixels i have a label $y_i = 1$, else $y_i = 0$. See figure 1 for a positive example, where the pneumothorax area is highlighted.

3.2 Data preprocessing

The dataset is unbalanced, with 72% of images not having the disease. To speed up computations and be able to do more experiments, we focus on a down-sampled and balanced dataset of a lower resolution. To be more specific, we use images of size 256×256, and 2669 samples of each class, resulting in a 5338 images dataset. We use half of it for training, and report performance on the other half.

[1] Available at `https://www.kaggle.com/c/siim-acr-pneumothorax-segmentation`

We are fully aware that the results can be substantially improved by increasing the resolution and using the full dataset, and do not intend to max out the scores, but rather to compare the relative changes of the proposed modifications on a strong baseline. Our choice allows for much faster run times and therefore makes it possible to conduct the experiments. With our hardware, a single run on the original data would have taken more than a day, instead of the $\approx$ 30 minutes needed here.

3.3 Baseline

Our baseline is derived from U-Net [16], a reference architecture for semantic segmentation, especially in medical applications. We schematize its architecture figure 2. U-Net has a encoder-decoder design, that are connected through a center block but also with intermediate skip connection. Our backbone is a pretrained ResNet-34 [10], itself a reference for image classification. We add a classification branch after the center block, which is weighted by a factor λ_{cls} in the total loss.

The U-Net decoder is based on *Convolution + ReLU + Batch Normalization* blocks, with a kernel size of 3. Each of its layer represented figure 2 is made of two of them, and we use the same structure for the center block. The global layer uses the same design, except for its first convolution which as a kernel size of 1, and it also contains a final convolution of 1 channel used to obtain the final mask. The decoder layers first applies up-sampling to the output of the previous layer, and concatenates it to the skip connection features.

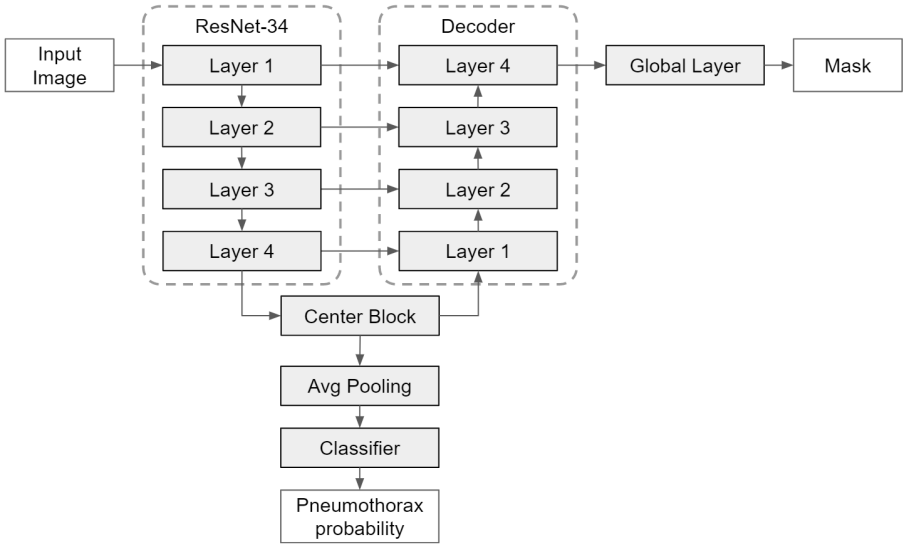


Fig. 2: The baseline U-Net architecture.

3.4 Training

The data are split in a stratified fashion into a training set and a validation set of equal size (2669 samples). We use the Rectified Adam optimizer [13] with a cosine linear annealing scheduler, proposed in [14]. We use the following training strategy which was experimentally determined. The encoder is pretrained [2] on ImageNet [6]. We first freeze the decoder, then jointly train everything. The most important hyperparameters are reported table 1. Augmentations were not looked into, although we still decided to apply horizontal flipping with a probability of 0.5.

	Parameter	Value
Global	Batch size	32
	Classification loss	BCE
	Segmentation loss	BCE
	λ_{cls}	0.1
Stage 1	Epochs	3
	Learning rate	5e-4
	Minimal learning rate	1e-4
Stage 2	Epochs	40
	Learning rate	5e-4
	Minimal learning rate	5e-5

Table 1: Main baseline hyperparameters

4 Proposed modifications

We study the impact of several methods to improve performances that we shall now briefly describe.

4.1 Lovász Loss

The Lovász-Softmax loss [2] acts as a surrogate to optimize the segmentation metric, the Intersection Over Union. As the latter is a discrete metric, the idea is to smoothly extend it to continuous values in $\mathbb{R}^+$. For that matter, the authors rely on *submodular* function analysis. A set function $\Delta : \{0,1\}^N \to \mathbb{R}$ is submodular if for all $\mathbf{A}, \mathbf{B} \in \{0,1\}^N$:

$$\Delta(\mathbf{A}) + \Delta(\mathbf{B}) \geq \Delta(\mathbf{A} \cup \mathbf{B}) + \Delta(\mathbf{A} \cap \mathbf{B})$$

Indeed, the IoU for a given class as a function mapping sets of pixels $S \subseteq \{0,1\}^N$ to $[0,1]$ is submodular. A natural way to extend it to $\mathbb{R}^+$ is to consider its

[2] Weights : `https://download.pytorch.org/models/resnet34-333f7ec4.pth`

convex closure. Computing a convex closure is generally NP-Hard, but hopefully for submodular functions, using the *Lovász extension* [1], it can be computed in polynomial time, and only involve differentiable operations. The Lovász-Softmax loss is obtained after combining it with the softmax and averaging over all classes.

We weighted the classification loss with $\lambda_{cls} = 0.1$ for the binary cross-entropy (BCE) and $\lambda_{cls} = 0.01$ when using the Lovász loss. We report table 2 the results of this modification on our baseline. It leads to a relative improvement of not only on the Dice (+0.85%) but also on the accuracy (+0.48%). This noticeable improvement is however around twice lower than the paper results on a general image segmentation benchmark.

Table 2: Influence of using Lovász and BCE losses with the baseline architecture.

Setup	Dice	Accuracy
Baseline (BCE)	0.583	0.828
Lovász	**0.588**	**0.832**

4.2 Atrous Spatial Pyramid Pooling (ASPP)

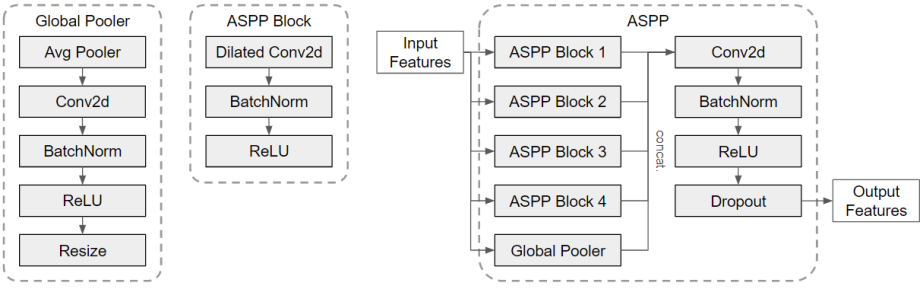


Fig. 3: Architecture of the ASPP module and its components.

ASPP (Atrous Spatial Pyramid Pooling) is one of the modules of the DeepLab [5] architecture. It is a computationally efficient way to deal with multi-scaled objects drawing inspiration from the idea of Pyramid Pooling. In our case, we use it to replace the Center Block (see fig. 2) of the model. ASPP expects as input features maps of size $H \times W \times C$. It applies 4 convolutions (of kernel size 1, 3, 3 and 3) with different dilatation coefficients (1, 6, 12 and 18) to it, as well as a pooling layer, such as depicted figure 3. The pooling layer averages the feature maps to a size of $c \times 1 \times 1$, and applies a convolution with a kernel size of 1. In this case, the convolution behaves like a fully connected layer. To

allow for the concatenation of all the outputs, the pooled output is resized to be the same size as the ASPP block outputs. All the convolutions are chosen to have the same number $n = 256$ of filters. After concatenation, we use a last convolution of kernel size 1 as well. According to its authors, ASPP aims at robustly segmenting at multiple scales. Indeed, using convolutions with different dilatation coefficients allows for receptive fields to be of different sizes.

Results are reported table 3 and using ASPP allows for a performance boost of $\approx 1\%$ on both metrics. In addition, the ASPP module allows for the use of a lighter model (≈ 800000 less parameters than the baseline one), although it did not reduce run times.

Table 3: Influence of using ASPP as the center U-Net block.

Setup	Dice	Accuracy
Baseline	0.583	0.828
ASPP	**0.592**	**0.838**

4.3 Hypercolumns

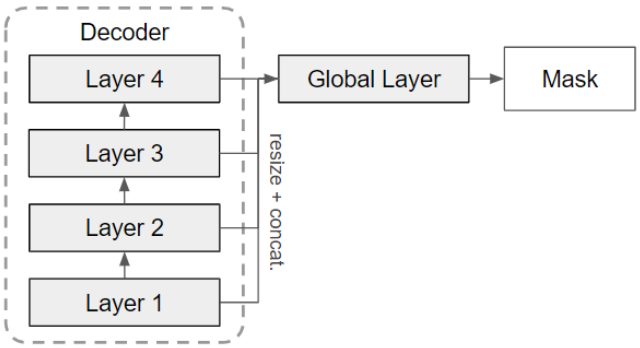


Fig. 4: Adding hypercolumns to the decoder.

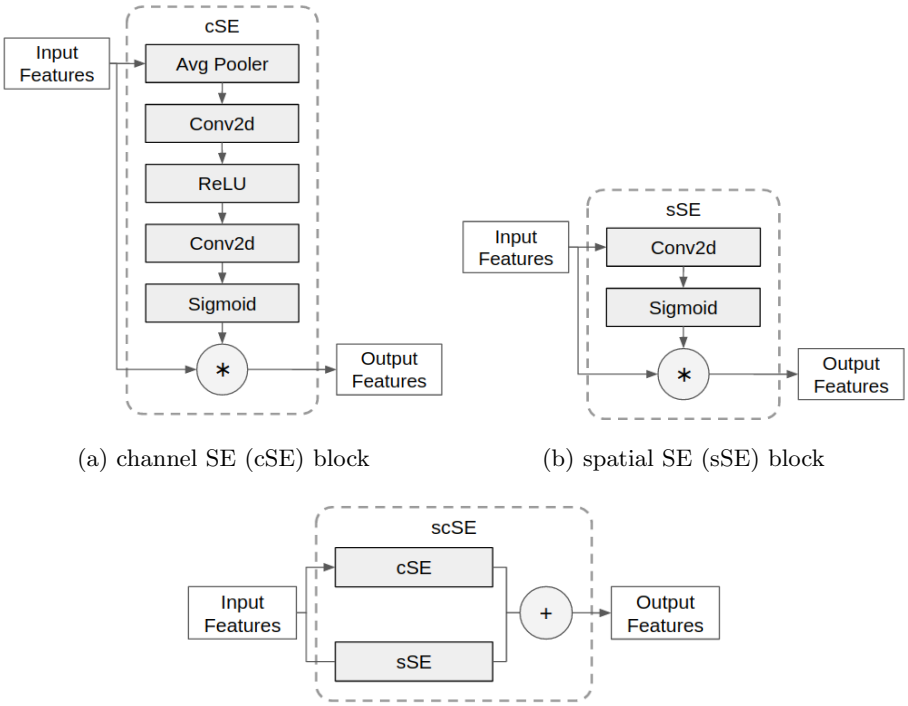
Introduced in [9], hypercolumns aim at combining the information of the outputs of all the layers of the decoder. To do so, the authors concatenate each output on the channel axis. For the concatenation to be possible, the output of the first three layers are up-sampled to be the same size as the one of the fourth. The concatenation is the fed to the global layer, of which the input size has to be updated consequently. The architecture change is schematized figure 4.

We report table 4 the influence of adding the hypercolumns to our model. Overall, we observe a small boost in performances for the segmentation task (+0.2%), and a quite big one for the classification task (+1.4%).

Table 4: Influence of using hypercolumns in the U-Net architecture.

Setup	Dice	Accuracy
Baseline	0.583	0.828
Hypercolumns	**0.585**	**0.842**

4.4 Spatial and channel Squeeze & Excitation (scSE)



(a) channel SE (cSE) block

(b) spatial SE (sSE) block

(c) spatial & channel SE (scSE) block

Fig. 5: Architecture of the scSE block and its components.

Spatial and channel Squeeze & Excitation (scSE) [17] draws inspiration from Squeeze and Excitation Networks [11]. It proposes an improvement to the con-

cept of channel Squeeze and Excitation blocks – cSE blocks – for semantic segmentation, and demonstrates a large improvement (+4-9% of Dice) on three medical applications. A scSE block combines a spatial recalibration (sSE) and a channel recalibration (cSE). All these blocks operate on an intermediate feature map $\mathbf{U}$ of size $H \times W \times C$. The cSE block compresses spatially with an average pooling the feature map $\mathbf{U}$ into z of size $1 \times 1 \times C$. The vector z goes through a MLP and a sigmoid activation, and the resulting vector $\hat{z} \in [0,1]^C$ reweights $\mathbf{U}$ channel-wise. Figure 5a summarizes the block. The spacial SE block, schematized figure 5b, spatially recalibrates $\mathbf{U}$. A convolution with one channel compresses channel-wise $\mathbf{U}$ into a feature map of size $H \times W \times 1$. Again, a sigmoid is applied and $\mathbf{U}$ is reweighted pixel-wise. Finally, the scSE block (see figure 5c) combines the two recalibrated features maps by summing them.

Plugging scSE blocks leads to an improvement of accuracy (+1%) but a decrease of the Dice score (−0.5%). Compared to the large relative improvements claimed by the authors on several medical imaging dataset, it is rather a surprise. In our case, we apply scSE on the layers of the decoder, before the first convolution and after the last one.

Table 5: Influence of using scSE in the U-Net decoder.

Setup	Dice	Accuracy
Baseline	**0.583**	0.828
scSE	0.578	**0.836**

5 Discussion

5.1 Ablation study

We report table 6 the results achieved by using different parameter settings. The best accuracy achieved is 0.842 using hypercolumns, whereas the best dice is 0.593 by using hypercolumns and ASPP. Although using the lovasz loss improved our baseline, it did not work really work well with other modifications for the segmentation task. scSE seems to be the proposed improvement that worked the worse for this task. Overall, combining different modifications hurts the results and did not give that much improvement.

5.2 Conclusion

In the deep learning research community, improvements over now-established baselines such as U-Net are published at very fast pace. Firstly, it is important to validate these claims on various datasets and settings. In our experiments, the claimed improvements were always largely over-estimated. Secondly, the behavior of these sometimes seemingly orthogonal methods in combination is not

Table 6: Ablation study.

Lovász	ASPP	Hypercolumns	scSE	Dice	Accuracy
				0.583	0.828
✓				0.588	0.832
	✓			0.592	0.838
		✓		0.585	**0.842**
			✓	0.578	0.836
✓	✓			0.574	0.829
✓		✓		0.586	0.829
✓			✓	0.582	0.835
	✓	✓		**0.593**	0.837
	✓		✓	0.588	0.827
		✓	✓	0.579	0.836
✓	✓	✓		0.579	0.832
✓	✓		✓	0.585	0.836
✓		✓	✓	0.578	0.834
	✓	✓	✓	0.574	0.840
✓	✓	✓	✓	0.574	0.837

obvious, and often not studied. It can even hurt performances, as we observed in our experiments. A similar phenomenon was previously highlighted by the example of batch normalization and dropout, two widely-adopted regularization methods that work very well independently, yet degrade performances when used together [12]. Our work could be generalized to other datasets to further validate our results, and with additional computing resources would benefit being conducted on high resolution images.

References

1. Francis Bach et al. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends® in Machine Learning*, 6(2-3):145–373, 2013.
2. Maxim Berman and Matthew B. Blaschko. Optimization of the jaccard index for image segmentation with the lovász hinge. *CoRR*, abs/1705.08790, 2017.
3. Sema Candemir, Stefan Jaeger, Kannappan Palaniappan, Jonathan P. Musco, Rahul K. Singh, Zhiyun Xue, Alexandros Karargyris, Sameer Antani, George Thoma, and Clement J. McDonald. Lung Segmentation in Chest Radiographs Using Anatomical Atlases With Nonrigid Registration. *IEEE Transactions on Medical Imaging*, 33(2):577–590, February 2014.
4. Yuan-Hao Chan, Yong-Zhi Zeng, Hsien-Chu Wu, Ming-Chi Wu, and Hung-Min Sun. Effective Pneumothorax Detection for Chest X-Ray Images Using Local Binary Pattern and Support Vector Machine, 2018.
5. Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs, 2014.

6. 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.
7. Kunio Doi and Shigeru Sanada. Method and system for automatic detection of ribs and pneumothorax in digital chest radiographs, September 1997.
8. Ofer Geva, Gali Zimmerman-Moreno, Sivan Lieberman, Eli Konen, and Hayit Greenspan. Pneumothorax detection in chest radiographs using local and global texture signatures. In *Medical Imaging 2015: Computer-Aided Diagnosis*, volume 9414, page 94141P. International Society for Optics and Photonics, March 2015.
9. Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization, 2014.
10. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
11. Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2017.
12. Xiang Li, Shuo Chen, Xiaolin Hu, and Jian Yang. Understanding the disharmony between dropout and batch normalization by variance shift. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.
13. Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.
14. Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2016.
15. Guoting Luo, Zhiqin Liu, Qingfeng Wang, Qiyu Liu, Ying Zhou, Weiyun Xu, Jun Huang, Jie Fu, and Jie-Zhi Cheng. Fully Convolutional Multi-Scale ScSE-DenseNet for Automatic Pneumothorax Segmentation in Chest Radiographs. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1551–1555, November 2019. ISSN: null.
16. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.
17. Abhijit Guha Roy, Nassir Navab, and Christian Wachinger. Recalibrating fully convolutional networks with spatial and channel 'squeeze & excitation' blocks. *CoRR*, abs/1808.08127, 2018.

Appendix A: Segmentation Results for Individual Methods

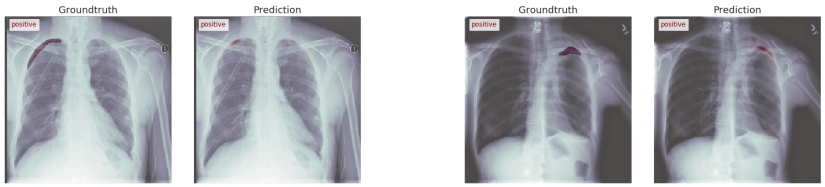


Fig. 6: Baseline

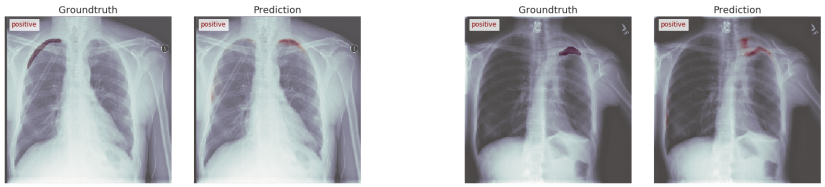


Fig. 7: ASPP

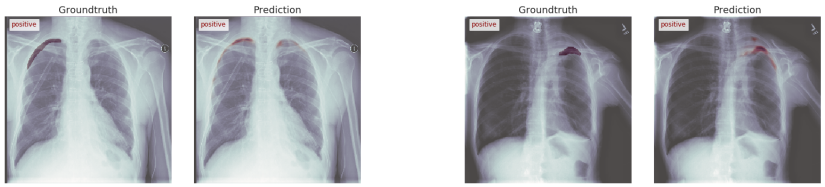


Fig. 8: Hypercolumns

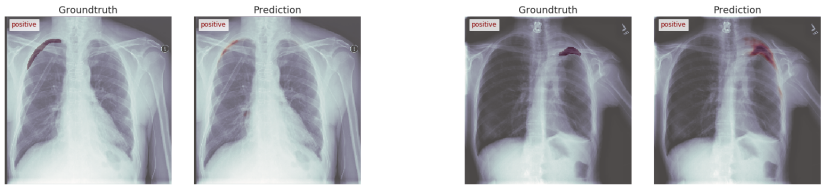


Fig. 9: Lovász loss

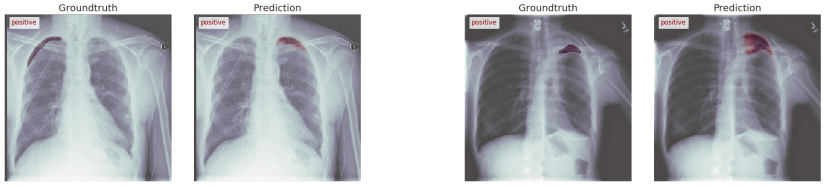


Fig. 10: scSE