# Approaching the semantic segmentation in medical problems – a pneumothorax solution

Author: Calin Timbus

# Agenda

- Own introduction
- Presentation
- Q&A + Further discussions

# About me – Professional part(1)

- Super passionate about ML/DL (everything related to, not only one domain/area)
- Currently doing my PhD in Mathematics (applications in Data Science)
- Currently Machine Learning engineer at Telenav.
- Teaching Introduction to Machine Learning lab at TUCN.
- Answering a lot on StackOverflow w.r.t ML/DL.

# About me – Personal part(2)

- I like playing basketball, visiting countries with lower temperatures :D.
- I am a sociable person – would wither in absence of interaction/communication with people.
- Enjoy learning foreign languages(even studying word etymology every so often)
- Love helping people: strongly related to teaching + StackOverflow part.

# Motivation for Pneumothorax (1)

- Pneumothorax is a condition in which a collapsed lung appears, mainly on account of accidents or shocks to the human body.

- Aiding tool for any radiographer, to easily see the location of pneumothorax; deep learning (A.I.) will never replace doctors.

- It can be seen on 2D data, 3D is not required, which is both harder and more expensive to acquire.

# Motivation for Pneumothorax (2)

- The medical domain should be the focus of the society for implementing A.I solutions; plenty are in dire need of support.

- In the era of Big Data, massive data acquisition should not be a problem...

- Nevertheless, gathering comprehensive and high-quality medical dataset is not a trivial task; on the contrary, much harder than for other simpler tasks (e.g. frog vs cat vs dog)

# Motivation for Pneumothorax (3)

- Young doctors may not have the necessary experience for easily detecting pneumothorax

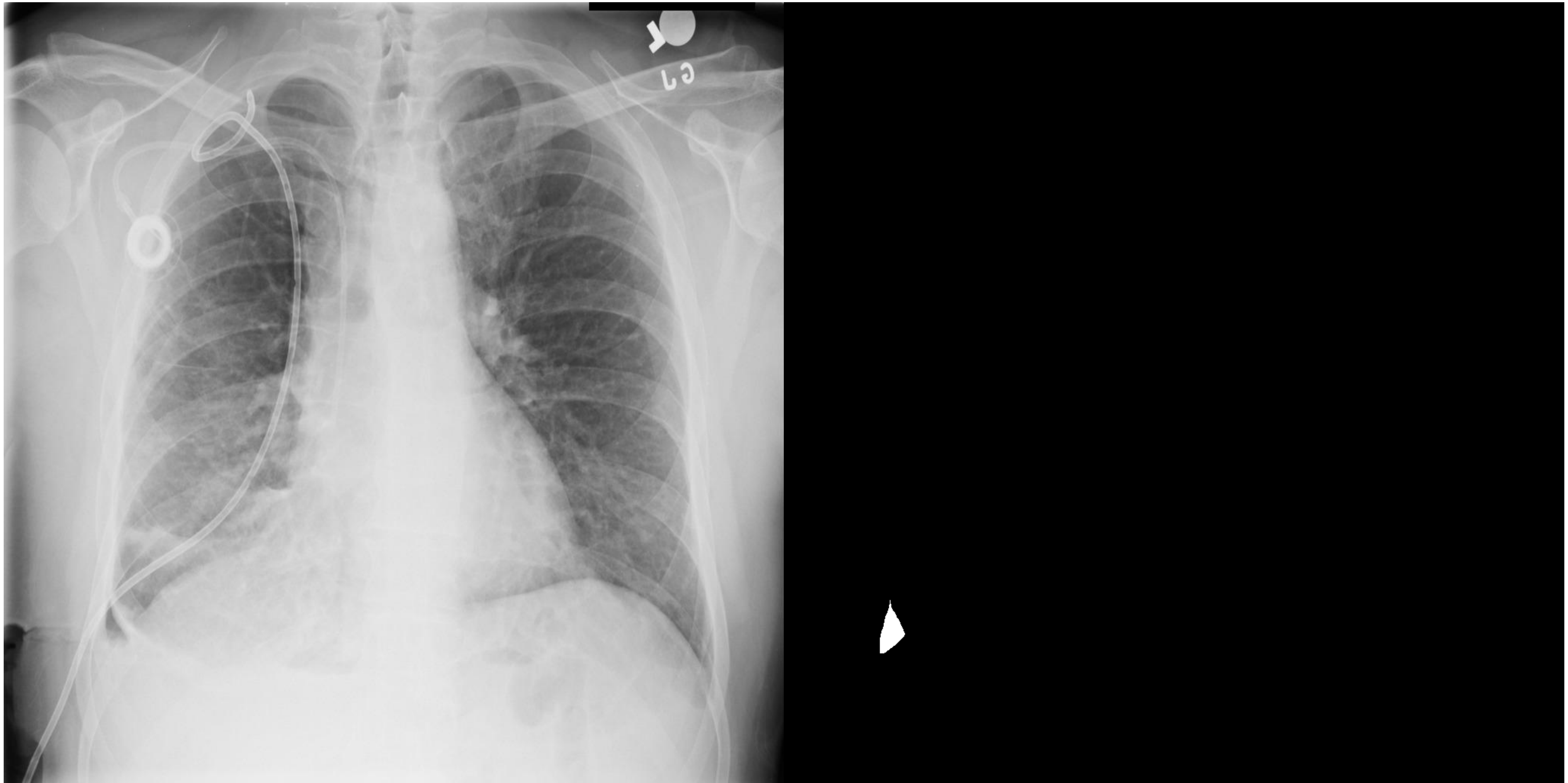
- Even for experienced doctors, detecting fine/smaller pneumothorax regions may prove to be a hard task

- Leveraging the power of deep learning given a consistent dataset may prove to offer a reliable and viable solution.

# Example of Pneumothorax (1)

- Left image contains pneumothorax
- The right belongs to a healthy patient
- Pneumothorax can come up in different ways : significant zone down to (a) negligible area(s).

# Example of Pneumothorax (2)



# Motivation for Deep Learning

- Image processing on based on deep learning has increased significantly in the recent period of time.
- Given a consistent dataset (dataset of high quality, correct labels, fine annotations) a deep learning approach can be implemented.
- Medical imaging datasets are in their infancy, although they have appeared recently even on Data Science competition platforms (**SIIM-ACR)**

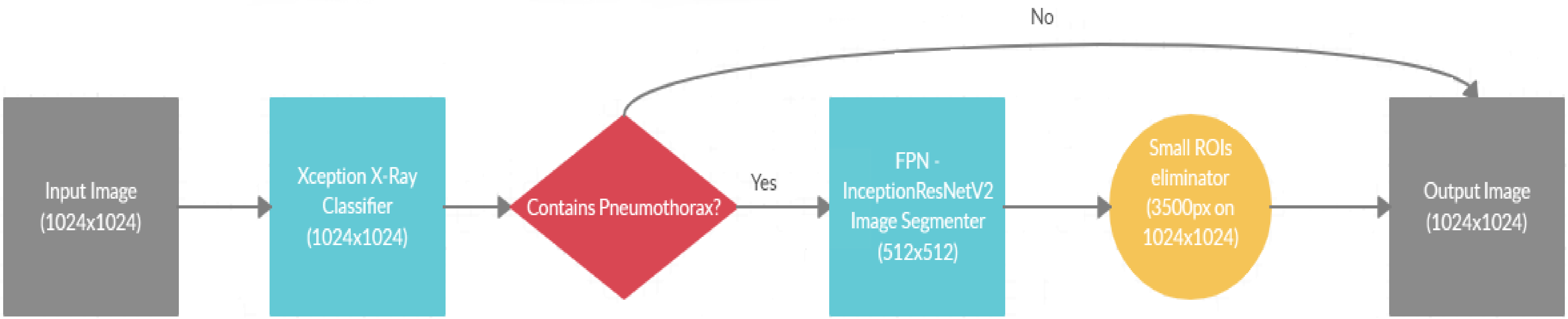
# Dataset Description

- Stage 1 and Stage 2 datasets (like many competitions on Kaggle)
- Stage 2 = Stage 1 * (Training Set + Test Set)
- Final results on Stage 2
- Almost 12.000 X-ray images with annotations (pneumothorax + non-pneumothorax)

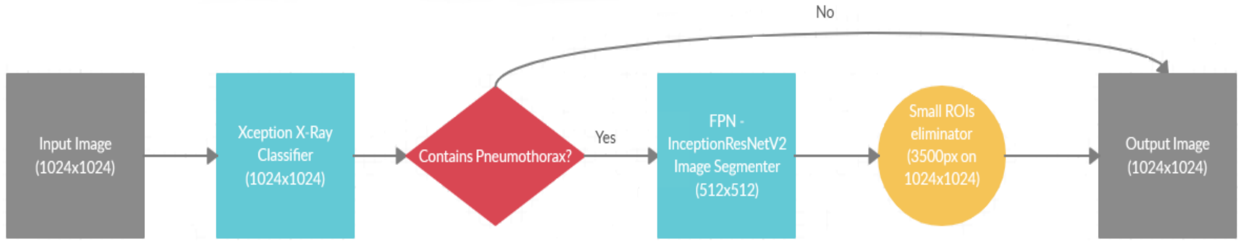
# Dataset Description - 2

- (1) The dataset is highly imbalanced (from a sample viewpoint) : 2883 images of pneumothorax vs 9378 images of non-pneumothorax).
- (2) The dataset is extremely imbalanced (from an region of interest viewpoint): less than 1% of the entire pixel distribution of the pneumothorax images contains in fact pneumothorax.
- Considering (1) and (2), a robust approach is needed to yield fruitful results (segment a pneumothorax image) == Exponential Imbalance.

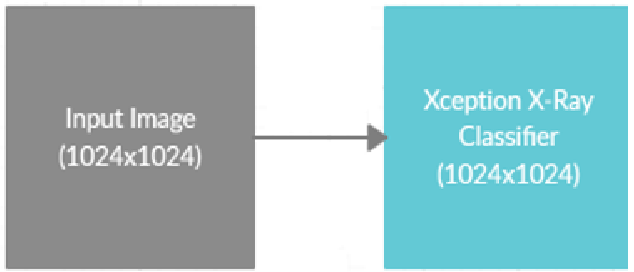
# Pipeline Proposed



# Classification Module - Rationale




- First tendency is to start using a segmentation network alone
- However, a lot of false positives may appear
- Only if a photo is pneumothorax should we try to segment it.
- Final results on Stage 2
- Almost 12.000 X-ray images with annotations (pneumothorax + non-pneumothorax)



# Classification Module - Details

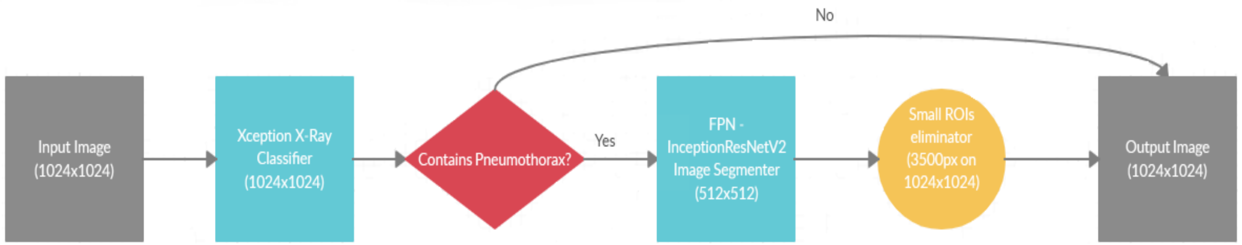
- Xception Network (Francois Chollet)
- Resolution : 1024x1024
- Started from pretrained weights
- Adam as an optimizer
- Aggressive augmentation (multiple augmentations (zoom-in,out,clahe) + high probability for augmentation 50%).
- No class weighting used (despite 3.5 imbalance factor)
- 20 epochs ( >= 20 results in overfitting)

# CLAHE in images

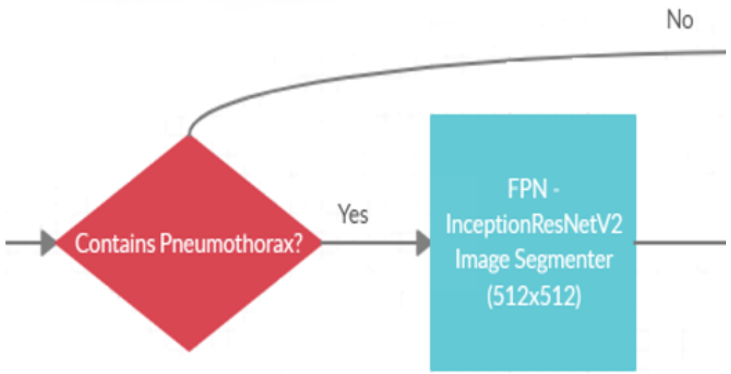
# Classification Module - Results

- Split 83.33%-16.66% (training-validation) (stratified split)
- MCC: 0.7702 (optimizing metric)
- Precision: 0.8010
- Recall: 0.8674
- Accuracy: 92.70%

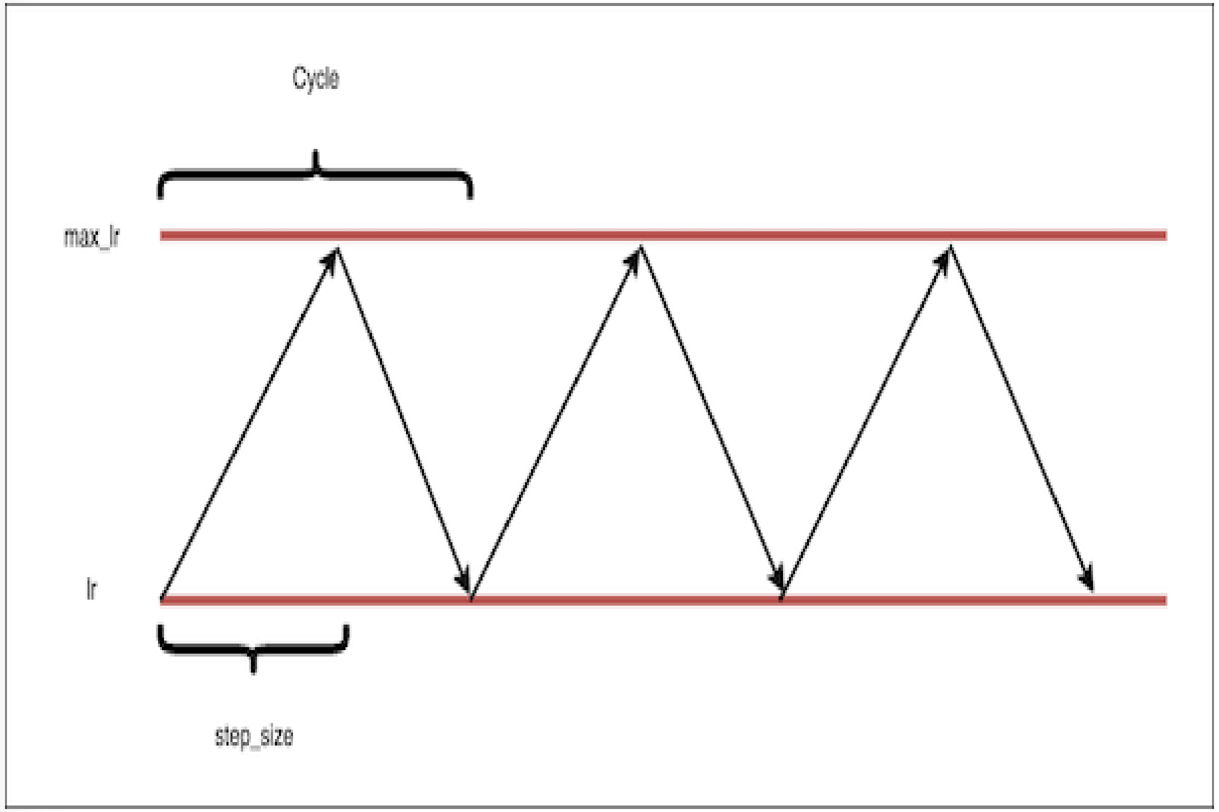
# Segmentation Module - Details

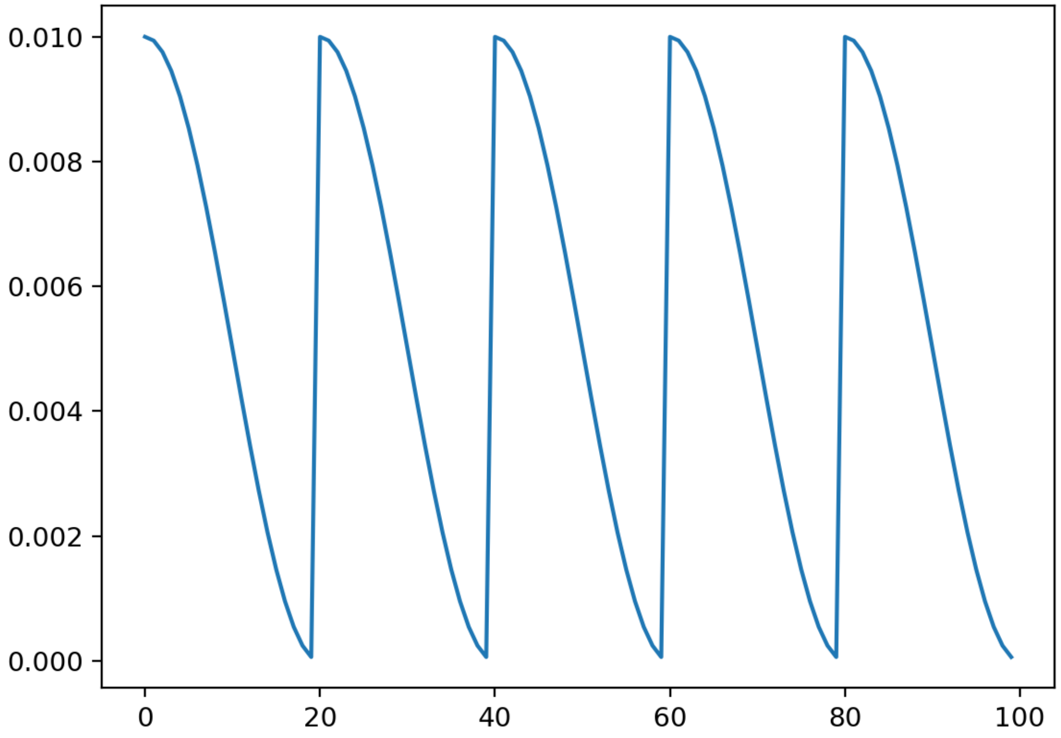



- InceptionResNetV2 as backbone (InceptionV4 + Residual Connections)
- FPN (Feature Pyramid Network)
- Dice Coefficient + CCE
- 2 epochs ½ frozen + 18 epochs fully trainable
- CosineAnnealingLearningRateScheduler + SWA (stochastic weight averaging (epochs 13-17))
- Adam as an optimizer



# Cyclical Learning Rate



# SWA –Detailed (1)

- The first model that stores the running average of model weights (w_swa in the formula). This will be the final model after the end of the training which will be used for predictions.
- The second model (w in the formula) that will be traversing the weight space, exploring it by using a cyclical learning rate schedule.

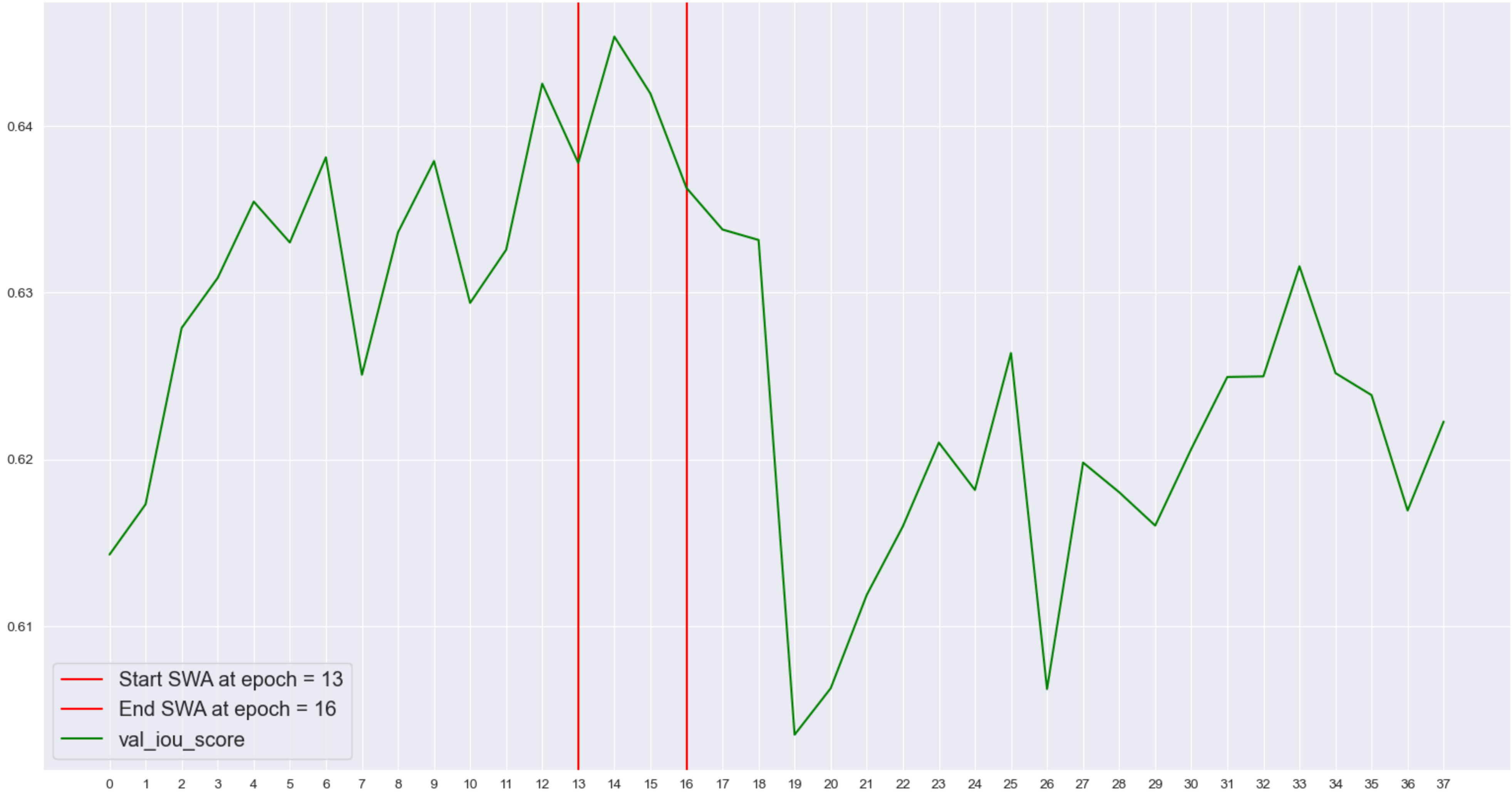
$$w_{\text{SWA}} \leftarrow \frac{w_{\text{SWA}} \cdot n_{\text{models}} + w}{n_{\text{models}} + 1,} \, ,$$

# SWA – Detailed (2)

- At the end of each learning rate cycle, the current weights of the second model will be used to update the weight of the running average model by taking weighted mean between the old running average weights and the new set of weights from the second model
- By following this approach, you only need to train one model, and store only two models in memory during training. For prediction, you only need the running average model.

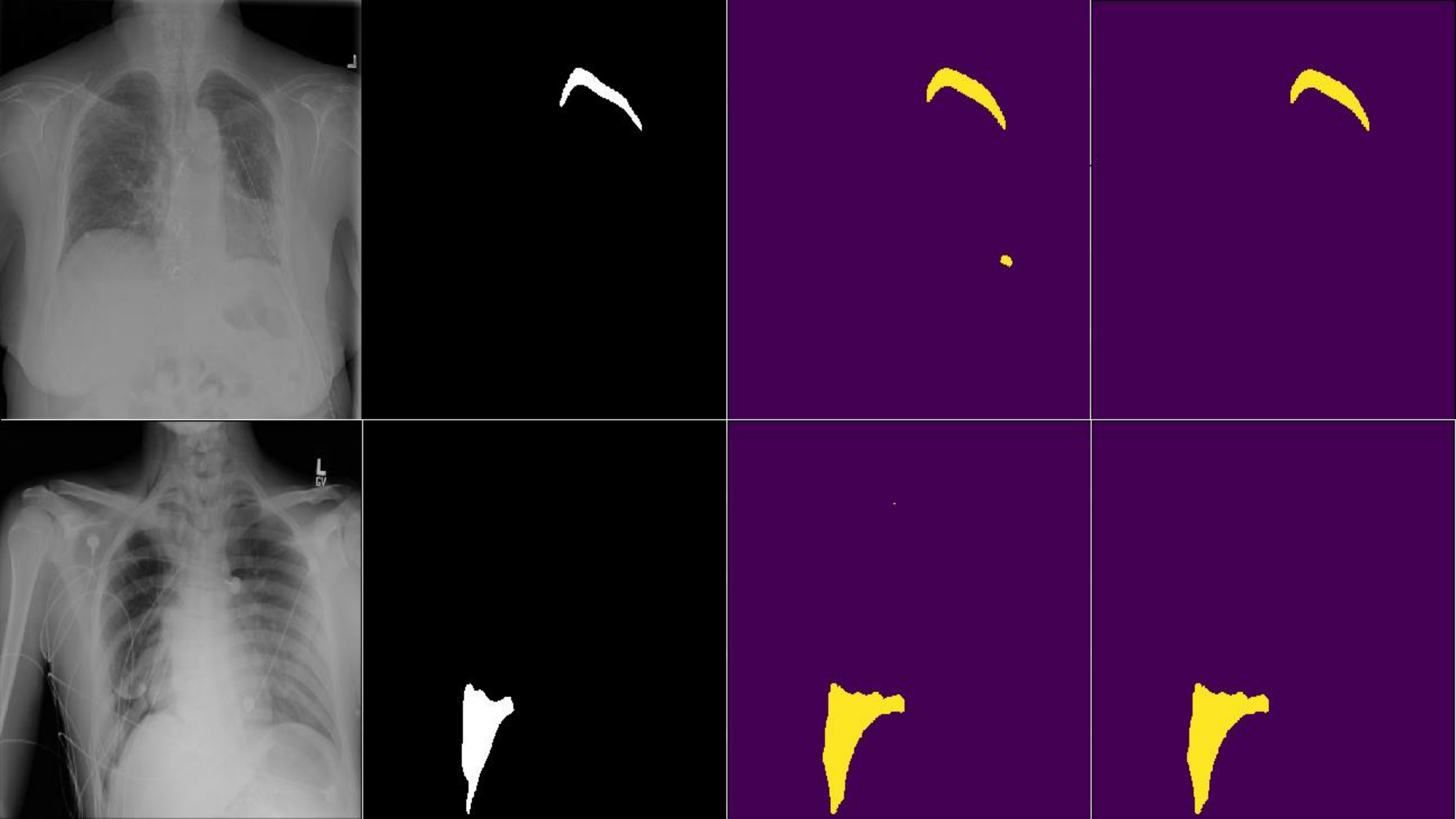
$$w_{\text{SWA}} \leftarrow \frac{w_{\text{SWA}} \cdot n_{\text{models}} + w}{n_{\text{models}} + 1,} \; ,$$

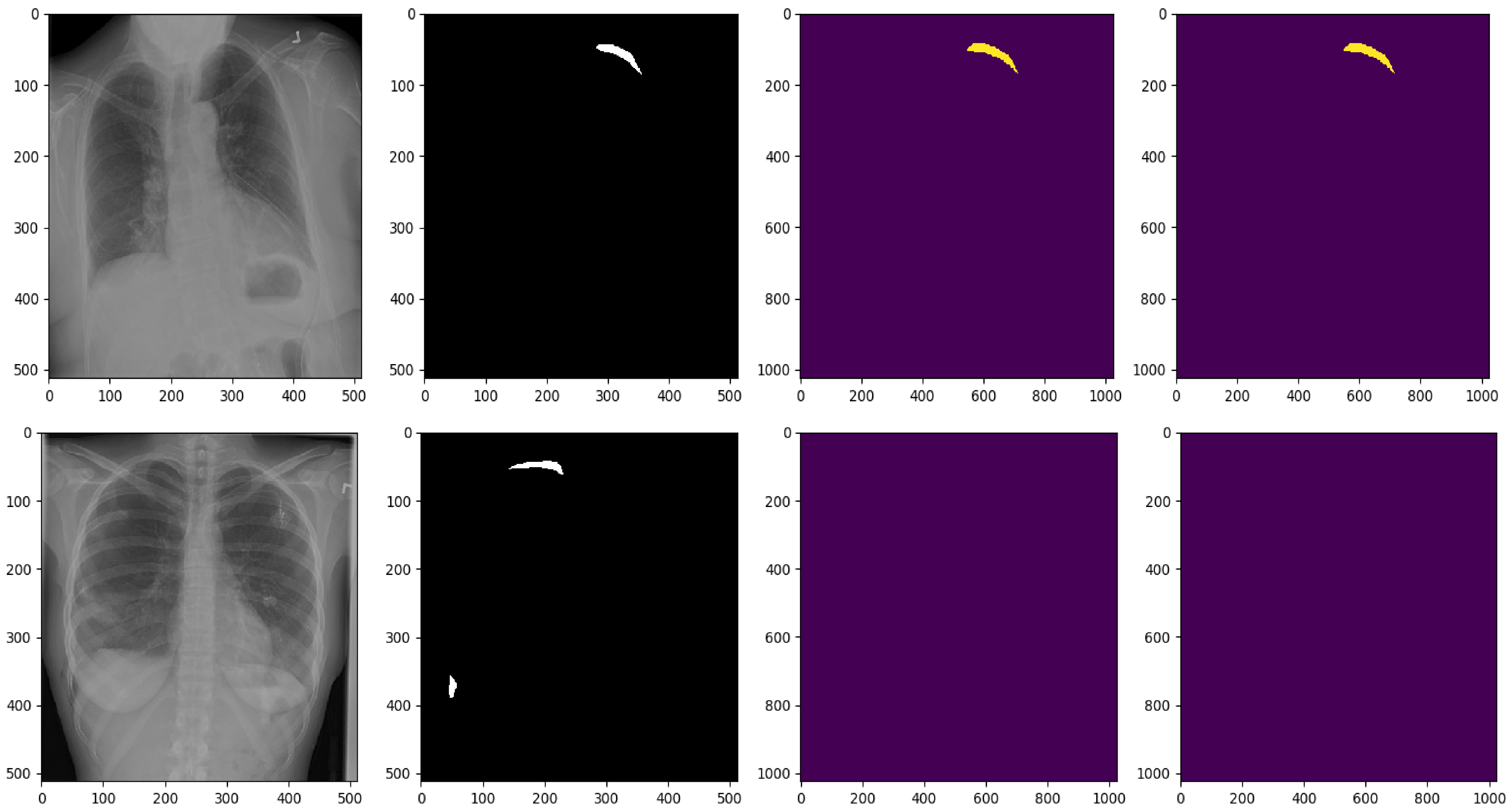
# SWA application



# Segmentation Module – Metric Results

- Split 83.33%-16.66% (training-validation) (normal split)
- IoU: 0.6332 (optimizing metric)
- F1-Score: 0.6773
- F2-Score : 0.6811
- The following slides will contain information about qualitative results


L
GV


0
100
200
300
400
500
0
200
400
600
800
1000

# Segmentation refinement

- In all of the previous slides, on the third column there is the result without small components removal.
- The fourth column contains the final result upon which the RLE is performed and sent to the competition.
- 1024x1024 3500 CC size for removal
- RLE is applied on 1024x1024 and represents the final result

# Additional observations

- 70.16 → 83.23 (simple segmentation vs pipeline)
- 83.23 → 83.47 (SWA ON)

# Conclusion

- 130th place out of 1476 in the competition (bronze place in the on-going competition)
- Thank you for paying attention.
- Q&A