

## Literature Review: Machine Learning Approaches to Diagnosing Pulmonary Emboli in CT Pulmonary Angiography Scans

### Introduction

A pulmonary embolism (PE) is a life-threatening acute obstruction of the pulmonary artery. Typically, this obstruction is caused by a thrombus (blood clot) formed in a deep vein of the leg, a condition known as deep vein thrombosis (DVT); although, it is also possible for the obstruction to originate from a fracture-induced fat embolism; tumor; or air bubbles (Chen et al., 2020; Cushman et al., 2004; Stringer et al., 2020; Wilson et al., 2020). This obstruction becomes lodged in the pulmonary artery, which causes a partial or complete blockage of blood flow resulting in decreased oxygenation and subsequent damage to vital organs. Rapid and accurate diagnosis can be complicated by the heterogeneity of presenting symptoms, ranging from difficulty breathing to circulatory collapse (Stein & Henry, 1997). Regardless of initial presentation, PE can rapidly lead to further complications, including hemodynamic instability, shock, and cardiac arrest (Obi & Packer, 2019; Wood, 2002). It is therefore imperative to promptly diagnose PE and begin administering treatment to improve patient outcomes.

### Deep Learning Approaches to Pulmonary Imaging Diagnosis

Ma et al. (2020) is a survey that summarizes the state-of-the-art of deep learning as applied to pulmonary imaging. Deep learning has been applied to aid diagnosis of pulmonary nodules, pneumonia, interstitial lung diseases, tuberculosis, and pulmonary embolism (PE). Focusing specifically on deep learning as applied to PE detection, I found only two papers of interest. Wen et al. (2019) uses deep learning to segment PE from lung images. Their neural network takes the CT scans as input and outputs a binary mask indicating location of embolism. We are unable to replicate this approach because we do not have the binary segmentation masks as part of training data. Our dataset only states the general location of the embolism and the severity.

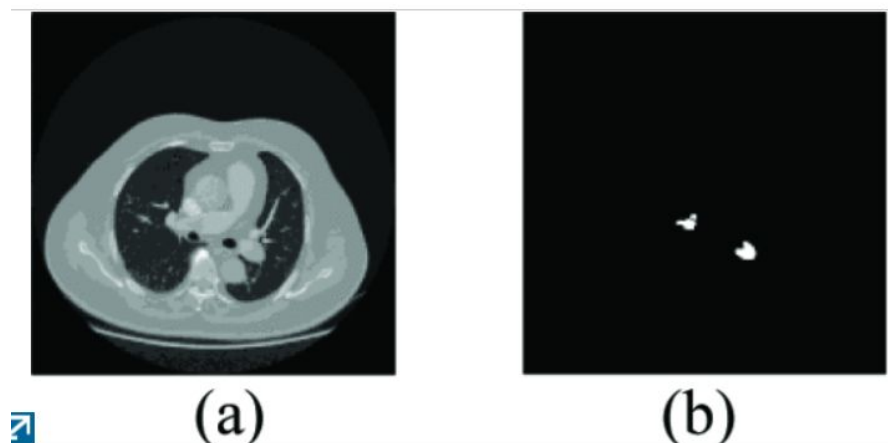


Image 1. Ma et al. Segmentation of PE.

Huang et al. (2020) implemented an open-source network called PENet that they showed to outperform state-of-the-art general purpose deep neural networks. From the paper, it is not clear what the output of the training data was. This paper gives us some insight into what parameters we may want to use for our implementation, and how to pre-process the CT data. The table below shows the performance comparison between PENet and general purpose networks (found on their github page).

	Internal dataset: Stanford	External dataset: Intermountain
Metric (AUROC) [95% CI]		
PENet kinetics pretrained	0.84 [0.82–0.87]	0.85 [0.81–0.88]
PENet no pretraining	0.69 [0.74–0.65]	0.62 [0.57–0.88]
ResNet3D-50 kinetics pretrained	0.78 [0.74–0.81]	0.77 [0.74–0.80]
ResNeXt3D-101 kinetics pretrained	0.80 [0.77–0.82]	0.83 [0.81–0.85]
DenseNet3D-121 kinetics pretrained	0.69 [0.64–0.73]	0.67 [0.63–0.71]

Image 2. (Huang et al.) <https://github.com/marshuang80/PENet>. NN Performance comparison.

The Github for PENet is relatively detailed. We could try to reproduce Huang et al.'s results using PENet on the Kaggle data. It would be interesting to see whether we find comparable accuracy. The authors have provided the trained weights for PENet, so we can use the network out-of-the-box. This will serve as a useful baseline.

Additionally, as seen in Image 2, ResNeXt3D (<https://github.com/facebookresearch/ResNeXt>.) has comparable performance to the PENet. Pretrained versions of ResNeXt3D are available and licensed under CC BY-NC 4.0. We may consider using this NN architecture for our model.

## Design Topics

Design decisions will mostly fall into two categories (1) Preprocessing and (2) Neural Network training.

### Pre-processing:

There is a wealth of information between the various Kaggle competitions that document how to pre-process Thoracic CT DICOMs. We can use

<https://www.kaggle.com/allunia/pulmonary-dicom-preprocessing> as a starting point.

Also useful (from IPF competition):

<https://www.kaggle.com/carlossouza/end-to-end-model-ct-scans-tabular> (Just the preprocessing classes, not the model)

1. Convert DICOM to numpy arrays

2. Transform to HU (Hounsfield Units) using the scaling factor and offset documented in the DICOM header.
  - a. Explanation of HU: [https://en.wikipedia.org/wiki/Hounsfield\\_scale](https://en.wikipedia.org/wiki/Hounsfield_scale)
  - b. CT voxels are values measured in HU (Hounsfield Units). This scale measures the radio-intensity. A smaller HU value corresponds to smaller attenuation of x-rays. Air does not attenuate x-rays well and has -1000 HU. Water has 0 HU. Our lungs are filled with air, which is why they appear dark on a CT scan.
    - i. Normal lung tissue is between -950 and -700 HU.
3. Resample DICOM images to same spacing (Interpolation). This is not trivial and we have to be careful not to blur the images when resampling (and consequently lose image data). Here are the relevant values in the DICOM header:
  - a. Z-spacing. (Calculate the difference in Z-position between adjacent slices)
  - b. X,Y-spacing.
  - c. Slice thickness.
  - d. Caveat: Some CT scans may not image the whole Z interval. For example, some CT scans will have slices spaced 1 cm apart; we don't get the data in between.
4. Lung segmentation: Remove everything that is not part of the lung from the CT scans. We may or may not want to do this.
  - a. Pros: May speed up training. NN does not waste time performing calculations on irrelevant anatomy.
  - b. Cons: If segmentation is not conservative enough, useful information may be segmented out. The NN should be capable of deciding where the useful information is.
  - c. Watershed technique (See <https://www.kaggle.com/carlossouza/osic-autoencoder-training> and <https://www.kaggle.com/aadhavvignesh/lung-segmentation-by-marker-controlled-watershed>)
  - d. Simple technique using connected region analysis. Basic idea is that all pixels under -700 is air (either air in the lungs or air outside the person). Preserve the largest two bodies of air that are inside the patient. This is very fast, but may not always work. Sometimes lungs may be connected. This does not work if the slices are not contiguous(if the slices are taken at large intervals).
    - i. Threshold the image to -700 to produce a binary image.
    - ii. Label the image based on connected component analysis (assign a different integer label to each connected region).
    - iii. Take the largest two connected components that are not connected to the boundaries of the image.
    - iv. Erode for conservative mask.
5. Image clipping(cropping) and resizing.
  - a. Crop out useless information (i.e. air outside the patient's body).

- b. Resize image to a certain size (e.g. 256x256). A smaller image size would make training the network more time-efficient.

## 6. Zero centering.

### **Data Augmentation/Regularization**

Deep Neural Networks are very susceptible to over-fitting. Over-fitting describes a model that predicts well using training data but does not predict well using testing data. Over-fitting results in a model that is not generalizable and therefore not useful in clinical settings. We need to deliver a model that generalizes well to data the network has not yet seen. In other words, we need the model to be robust.

There are a few classic ways to achieve this:

- Huang et al. use random cropping, rotation of up to 15 degrees, and randomly shifting z-coordinate of slices.
- Some other methods: adding pixel-wise gaussian noise, flipping the image.

The goal is to make the network robust to variations in the data. An image with PE is still an image with PE if we rotate it, flip it, shift it, or add some noise.

### **Pre-Training**

Tajbakhsh et al. (2016) provide a general discussion of the merits of pre-training a NN. Pre-training is often useful because it initializes the weights (parameters) of the NN to reasonable values. This reduces the time it takes to train the NN to a specific application. In simpler terms: the goal of training a NN is to minimize a loss function with respect to a parameter space. From scratch, it takes many training iterations to converge on a minimum. Using a pre-trained NN, it will take fewer training iterations to converge, because the initial state is already close to the minimum.

As seen in Huang et al., we will likely use a pre-trained model. Pre-trained models are usually trained on generic non-medical data. Huang et al.'s PENet was pre-trained on Kinetics-600 dataset. Kinetics-600 dataset is a set of video clips of human actions (e.g. hugging, playing an instrument). Note that video data is used because PENet is a 3D network, and video data is 3D.

### **Evaluation**

The metric used in the Kaggle competition to evaluate the algorithm is weighted log loss, where the weights reflect relative importance of certain labels. For this evaluation, there is exam-level weighted log loss and image-level weighted log loss. See below for a table for  $w_j$  (weights for label  $j$ ) and an explanation of exam-level weighted log loss.

Label	Weight
Negative for PE	0.0736196319
Indeterminate	0.09202453988
Chronic	0.1042944785
Acute & Chronic	0.1042944785
Central PE	0.1877300613
Left PE	0.06257668712
Right PE	0.06257668712
RV/LV Ratio >= 1	0.2346625767
RV/LV Ratio < 1	0.0782208589

*Image 3. Weights per the Kaggle competition evaluation metric. [Source: <https://www.kaggle.com/c/rsna-str-pulmonary-embolism-detection/overview/evaluation>]*

Variables are summarized below:

Variable	Range of Values	Representation
i	1,2,3,...N	Exams in test set
j	1,2,3,...9	Labels
$w_j$	See table above	Weights for label j
$y_{ij}$	1 or 0	1 if label j is annotated to exam i; 0 otherwise
$p_{ij}$	[0,1]	Predicted probability that $y_{ij} = 1$

The binary weighted log loss function

$$L_{ij} = -w_j * [y_{ij} * \log(p_{ij}) + (1 - y_{ij}) * \log(1 - p_{ij})]$$

is used for each label, then the mean log loss of all labels is calculated.

The image-level weighted log loss function is associated with the following variables:

Variable	Range of Values	Representation
w	0.07361963	Weight
i	1,2,3...N	Exams
$n_i$	variable	Number of images in exam i
k	1,2,... $n_i$	Image number
$y_{ik}$	1 or 0	1 if image k in exam i has 'PE present on image' annotation; 0 otherwise
$p_{ik}$	[0,1]	Probability that $y_{ik} = 1$
$q_i$	[0,1]	Proportion of positive images in exam i

The image-level log loss function is

$$L_{ik} = -[w * q_i] * [y_{ik} * \log(p_{ik}) + (1 - y_{ik}) * \log(1 - p_{ik})]$$

The total loss is given by the average of all image- and exam-level loss divided by all image- and exam-level weights.

## References

- Chen, Z., Feng, T., Wang, M., Xu, X., Wang, Y., Li, Y., & Min, L. (2020). Rare cause of repeated pulmonary embolism: A case of primary pleural squamous cell carcinoma and literature review. In *BMC Pulmonary Medicine* (Vol. 20, Issue 1).  
<https://doi.org/10.1186/s12890-020-1077-2>
- Cushman, M., Tsai, A. W., White, R. H., Heckbert, S. R., Rosamond, W. D., Enright, P., & Folsom, A. R. (2004). Deep Vein Thrombosis and Pulmonary Embolism in Two Cohorts: The Longitudinal Investigation of Thromboembolism Etiology. *Am J Med*, 117, 19–25.  
<https://doi.org/10.1016/j.amjmed.2004.01.018>
- Huang, S. C., Kothari, T., Banerjee, I., Chute, C., Ball, R. L., Borus, N., Huang, A., Patel, B. N., Rajpurkar, P., Irvin, J., Dunnmon, J., Bledsoe, J., Shpanskaya, K., Dhaliwal, A., Zamanian, R., Ng, A. Y., & Lungren, M. P. (2020). PENet—a scalable deep-learning model for automated diagnosis of pulmonary embolism using volumetric CT imaging. *Npj Digital Medicine*, 3(1). <https://doi.org/10.1038/s41746-020-0266-y>
- Ma, J., Song, Y., Tian, X., Hua, Y., Zhang, R., & Wu, J. (2020). Survey on deep learning for pulmonary medical imaging. In *Frontiers of Medicine* (Vol. 14, Issue 4, pp. 450–469).  
<https://doi.org/10.1007/s11684-019-0726-4>
- Obi, M., & Packer, C. D. (2019). Submassive Pulmonary Embolism: A Re-evaluation of Hemodynamic Instability. *Cureus*. <https://doi.org/10.7759/cureus.4644>
- Stein, P. D., & Henry, J. W. (1997). Clinical characteristics of patients with acute pulmonary embolism stratified according to their presenting syndromes. *Chest*, 112(4), 974–979.  
<https://doi.org/10.1378/chest.112.4.974>
- Stringer, B., Henry, L., & Foley, R. (2020). Iatrogenic air embolism. In *Clinical Case Reports* (Vol. 8, Issue 9, pp. 1850–1851). <https://doi.org/10.1002/ccr3.3007>
- Tajbakhsh, N., Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B., & Liang, J. (2016). Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? *IEEE Transactions on Medical Imaging*, 35(5), 1299–1312.  
<https://doi.org/10.1109/TMI.2016.2535302>
- Z. Wen, H. Wang, H. Yuan, M. Liu and X. Guo, "A method of pulmonary embolism segmentation from CTPA images based on U-net," *2019 IEEE 2nd International Conference on Computer and Communication Engineering Technology (CCET)*, Beijing, China, 2019, pp. 31-35, doi: 10.1109/CCET48361.2019.8989357.
- Wilson, A., Hanandeh, A., Shamia, A. A., Louie, K., & Donaldson, B. (2020). Effective

Management of Femur Fracture Using Damage Control Orthopedics Following Fat Embolism Syndrome. Cureus. <https://doi.org/10.7759/cureus.7455>

Wood, K. E. (2002). Major pulmonary embolism: Review of a pathophysiologic approach to the golden hour of hemodynamically significant pulmonary embolism. In *Chest* (Vol. 121, Issue 3 SUPPL., pp. 877–905). <https://doi.org/10.1378/chest.121.3.877>