PDE-based Neural Networks for Lung Cancer Detection using 3-D LDCT Images

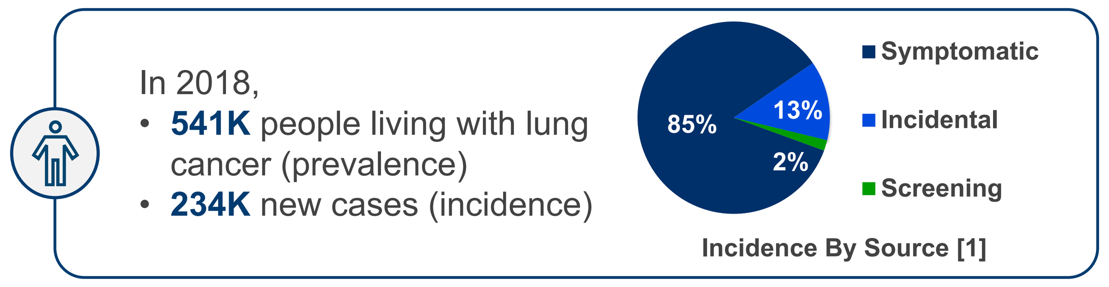
Derek Onken
PhD Candidate, Department of Computer Science, Emory University
Data Science Intern, UnitedHealth Group R&D, Summer 2019

Motivation

Lung cancer:

- is responsible for the most deaths of all cancers
- begins localized in the lungs and spreads

  5-year survival rates:
  - 19% (all stages)
  - 56% (still localized to the lungs)
- can be caught and treated early, reducing mortality



Annual screening is recommended for the non-symptomatic high-risk population (smokers with >15 pack-years; age 55-74). Radiologists read these scans to determine cancer diagnosis.

	Actual True	Actual False	Total
Predicted True	270 TP	6,911 FP	7,181
Predicted False	18 FN	19,043 TN	19,061
Total	288	25,954	26,242

$$\text{Sensitivity}\left(\frac{TP}{TP+FN}\right) = 93.8\%$$
$$\text{PPV}\left(\frac{TP}{TP+FP}\right) = 3.8\%$$
$$F_1\left(\frac{2TP}{2TP+FN+FP}\right) = 7.2\%$$

Data from LDCT group baseline (T0) of Nat'l Lung Cancer Screening Trial (NLST) [2]

Current process can be expensive for providers, and patients experience delays and high costs.

Commercial Population [3]

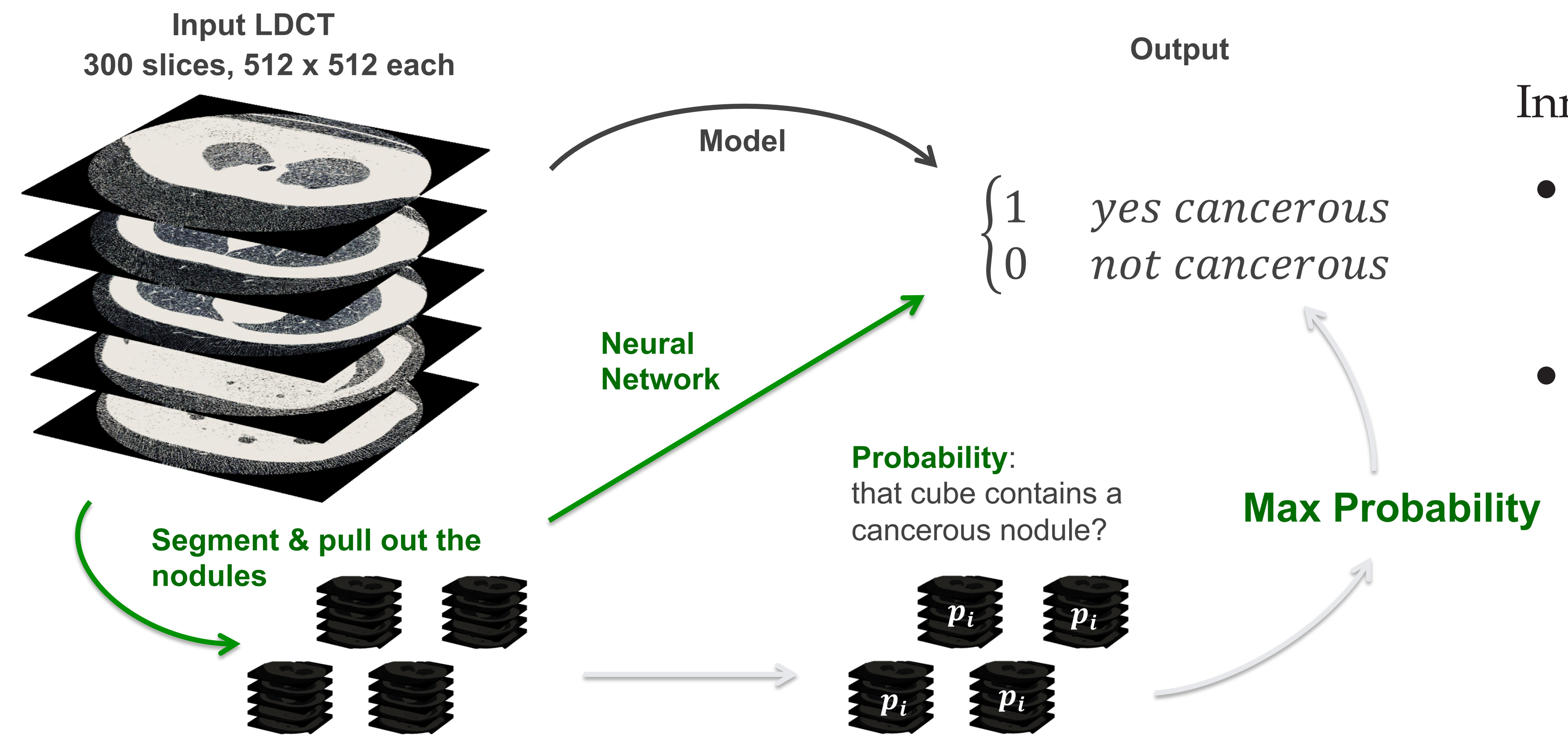
Of adult patients diagnosed with non-small cell lung cancer (2007-2011)
- **~94%** experienced a delay of 5-6 months
- Mean per patient per month in total health care costs was **\$2,407 ± \$3,364**

False positives render many of these costs wasteful.

⇒ **Reduction in false positives at initial screening saves patients and providers time and money.**

Problem & Approach

Develop a model that classifies a 3-D low-dose computed tomography (LDCT) scan as cancerous or non-cancerous. **Performance goal**: fewer false positives than practicing radiologists while matching sensitivity.



Inner workings of the model:

- Segmenter pulls out smaller cubes with the most likely cancerous nodules (for scalability)
- PDE-based Classifier
  - Predict cancer likelihood for each cube
  - Use from the max likelihood cube for the patient

Optimal Control in NNs

Goal: Find a function $f : \mathbb{R}^p \times \mathbb{R}^n \to \mathbb{R}^m$ and parameters $\boldsymbol{\theta} \in \mathbb{R}^p$ such that $f(\boldsymbol{\theta}, \mathbf{y}_0) \approx \mathbf{c}$ for every training input $\mathbf{y}_0 \in \mathbb{R}^n$ and its label $\mathbf{c} \in \mathbb{R}^m$.

We make this continuous, viewing weights $\boldsymbol{\theta}$ and features $\mathbf{y}$ as functions of time $t \in [0, T]$.

The neural network $f(\boldsymbol{\theta}, \mathbf{y}_0) = \mathbf{y}(T)$ performs a nonlinear transformation of inputs $\mathbf{y}_0$ satisfying the ordinary differential equation (ODE),

$$\begin{aligned} \partial_t \mathbf{y}(t) &= \ell(\boldsymbol{\theta}(t), \mathbf{y}(t)), \quad \text{for} \quad t \in (0, T] \\ \mathbf{y}(0) &= \mathbf{y}_0, \end{aligned} \qquad (1)$$

where $\ell : \mathbb{R}^p \times \mathbb{R}^n \to \mathbb{R}^n$ is a neural network.

A forward Euler discretization [4] of Eq. (1) is the N-layer ResNet

$$\mathbf{y}_{j+1} = \mathbf{y}_j + h\,\ell(\boldsymbol{\theta}_j, \mathbf{y}_j), \quad \text{where } j = 0, 1, \ldots, N-1,$$

with step size $h = T/N$. Borrowing from optimal control, the $\boldsymbol{\theta}_j$ are *control layers* and the $\mathbf{y}_j$ are *state layers*.

Training: Tune $\boldsymbol{\theta}$ and linear layer $\mathbf{W}$ by solving

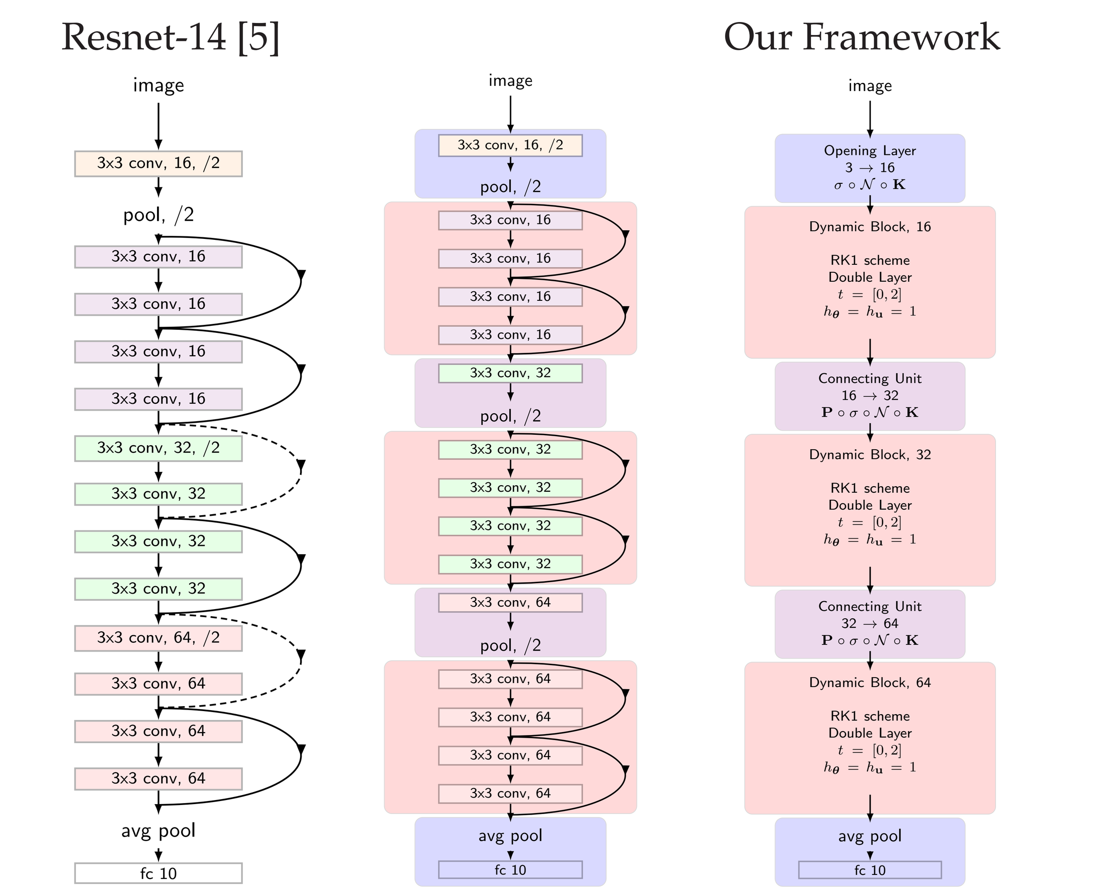
$$\min_{\boldsymbol{\theta}, \mathbf{W}} \frac{1}{s} \sum_{k=1}^{s} \text{loss}(f(\boldsymbol{\theta}, \mathbf{y}_0^k)\mathbf{W}, \mathbf{c}^k) + \text{regularizer}(\boldsymbol{\theta}, \mathbf{W})$$

across all s training inputs.

Decoupling Weights & Layers

Goal: Improve training by maintaining few parameters but with many layers

We develop a generalized ResNet for the classifier:



Fixed-width portions of ResNet = a Dynamic Block

Each block contains:

- continuous ODE like Eq. (1)
- ODE solver scheme (e.g., Runge-Kutta 4)
- discretization for the solver
- neural network layer ℓ

For ℓ, we experiment with:

- Double Layer: $\sigma_2 \circ \mathcal{N}_2 \circ K_{\boldsymbol{\theta}_2} \circ \sigma_1 \circ \mathcal{N}_1 \circ K_{\boldsymbol{\theta}_1}(\mathbf{y})$
- Double Symmetric Layer [6]: $-K_{\boldsymbol{\theta}}^{\top} \circ \sigma \circ \mathcal{N} \circ K_{\boldsymbol{\theta}}(\mathbf{y})$

for activation functions σ, normalizations $\mathcal{N}$, and convolution operators K defined by weights $\boldsymbol{\theta}$

Funding

Supported by the National Science Foundation awards DMS 1522599 and CAREER DMS 1751636, Binational Science Foundation Grant 2018209, NVIDIA Corporation, and UnitedHealth Group.

Collaborators

- Lars Ruthotto (Emory University)
- Rachel Jennings (UnitedHealth Group)
- Stephen Garth (UnitedHealth Group)
- Simion Novikov (Ben Gurion, Israel)
- Eran Treister (Ben Gurion, Israel)
- Eldad Haber (UBC, Vancouver)

Results

- Use pre-trained segmenter [7]
- Classifier uses Double Symmetric Layer
- Minimize Focal Loss
- Trained on a subset of 280 patient scans
- Requires 1 week to train on a GPU with 16GB RAM

	Radiologists (NLST patients)		Google AI [8] (NLST patients)		Ours: Training Set (NLST patients)		Ours: Validation Set (NLST patients)	
	Actual True	Actual False	Actual True	Actual False	Actual True	Actual False	Actual True	Actual False
Predicted True	270	6,911	82	1,260	6	13	2	10
Predicted False	18	19,043	4	5,370	1	260	0	133
% cancerous (actual)	1.1%		1.3%		2.5%		1.4%	

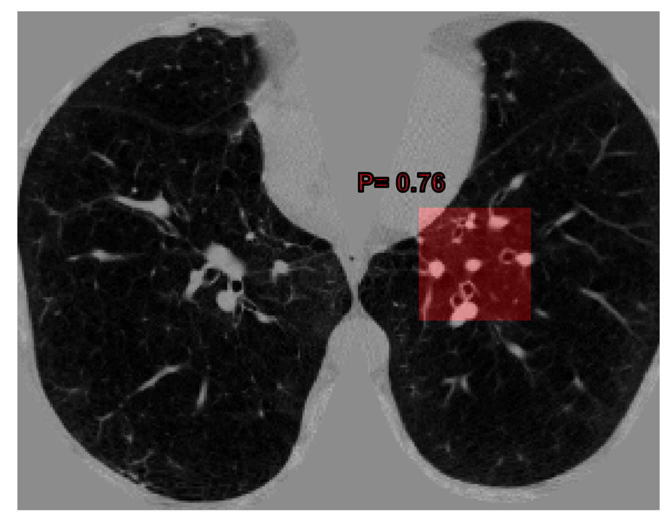
Radiologists	Google AI [1]	Metric	Training	Validation
0.94	**0.95**	**Sensitivity**	**0.86**	**1.00**
0.04	**0.06**	**PPV**	**0.32**	**0.17**
0.07	**0.11**	**F_1-score**	**0.46**	**0.29**
0.73	0.81	Specificity	0.95	0.93
0.99	0.99	NPV	0.99	1.00
0.74	0.81	Accuracy	0.95	0.93

Good results (but expensive) on small subset
Need to increase training data and class imbalance

Implementation in Practice

Goal: Provide physicians some interpretation of the model's output

- The segmenter provides region of interest cubes
- Each region has a cancer probability p_i
- Show those to the radiologist



Future Directions

- Scale current method to all 15,000 patient scans
- Apply PDE interpretation to the segmenter
- Predict 5 classes (Lung-RADS)
- Add a recurrent component to compare against past scans (nodule growth)

Software

PyTorch implementation for CIFAR-10 and STL-10 available on Emory's Machine Learning and Inverse Problems Github repository:

github.com/EmoryMLIP/DynamicBlocks

Lung-specific hyperparameters and data unavailable.

References

[1] Disease Inception Accelerator. *Patient Journey Research.* 2018.
[2] NLST Reseach Team. *Results of Initial Low-Dose Computed Tomographic Screening for Lung Cancer.* NEJM, 2013.
[3] T Gildea et al. *A Retrospective Analysis of Delays in Diagnosis.* ClinicoEconomics and Outcomes Research: CEOR, 2017.
[4] E Haber, L Ruthotto. *Stable Architectures for Deep Neural Networks.* Inverse Problems, 2017.
[5] K He et al. *Deep Residual Learning for Image Recognition.* IEEE CVPR, 2016.
[6] L Ruthotto, E Haber. *Deep Neural Networks Motivated by Partial Differential Equations.* JMIV, 2019.
[7] F Liao et al. *Evaluate the Malignancy of Pulmonary Nodules Using the 3-D Deep Leaky Noisy-or Network.* IEEE TNNLS, 2019.
[8] D Ardila et al. *End-to-end Lung Cancer Screening with 3-D Deep Learning.* Nature Med., 2019.