Predicting Lung Cancer Incidence from CT Imagery

CS 231N Final Project | Spring 2017

TA: Zhao (Joe) Chen

Darren Baker
Ali Chaudhry
Jen Kilpatrick
{drbaker, alikc, jdki}
@stanford.edu

Problem Background

- **Lung cancer** is one of the most common forms of cancer in the U.S.
  - Responsible for many **deaths** and significant **health care costs**
- Low-dose CT (computed tomography) scans are used by human radiologists to assess a patient's risk of lung cancer
  - **Radiologist's goal:** identify small **"nodules"** (tissue growths) in the lungs that often precede cancer
  - Challenge: nodules are small and difficult to classify, leading to **high false positive rates**
- **Our goal:** use image recognition techniques to better **predict cancer development** from CT scan data

Data Overview

- Our primary dataset consists of **3-dimensional CT images from ~1600 patients** at high risk of lung cancer
- Each patient's data contains between **100 and 500 axial (top-down) grayscale "slices"** of the lungs
- Data labels: **"1" if cancer diagnosed** within the next year, "0" otherwise

Modeling & Prediction Approaches

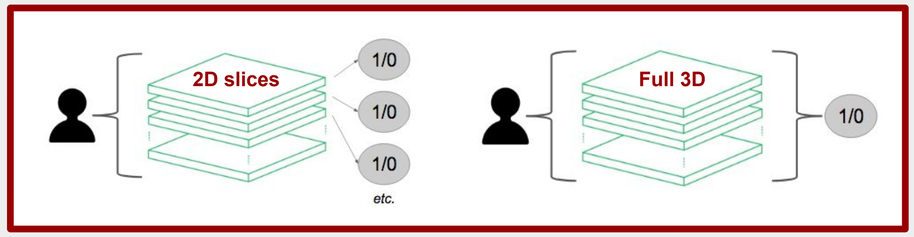
Data preparation

As with many medical imaging assets, our dataset required substantial **preprocessing** to prepare it for model training / prediction. Key steps:

- **Extracting** metadata & pixels from DICOM files to numpy format
- **Rescaling** pixel data into "Hounsman units" (standard for CT)
- **Compressing / sorting / saving** extracted data for quick loading

Convolutional models

We initially developed two convolutional models for cancer prediction:



2-dimensional model

- **At training time:** Considers each slice each independently, trains on overall patient label
- **At test time:** Classifies slices independently, then predicts label using threshold over slices
- **Architecture:** 1-3 2D conv layers (+ pooling/batch norm), 2 FC layers, softmax loss
- **Pros:** training speed; uniformity over input dimensions (512x512)
- **Cons:** "label spraying"; loss of 3D structure

3-dimensional model

- **At training / test time:** Considers 3D pixel matrix for each patient, and trains/evaluates on patient label
- **Architecture:** 2-5 3D conv layers (+ pooling/batch norm), 1-3 FC layers, softmax loss
- **Pros:** preserves 3D spatial relationships & integrity of patient label
- **Cons:** variable depths require padding/truncation; training slow and memory-constrained

Experimentation & tuning

We explored a range of **architectures** (layer counts, conv filter sizes, etc.) and **hyperparameter values** (slice count thresholds, learning rate, etc.) to try to optimize model performance.

Results

Our experiments show that 2D & 3D convolutional models have **not been able to learn much** on the raw pixel data.

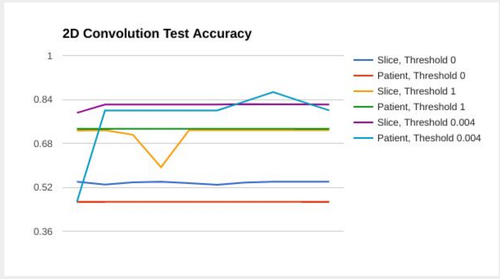
- Classification accuracy is only **slightly (2-8%) above the "naive" value** of predicting the majority class.
- 3D conv accuracy is somewhat **worse than 2D** -- perhaps due to padding/truncation.
- Model **training behavior is volatile** -- frequently jumps to predicting same label for all datapoints!

True distribution

	Train	Test
Cancer	25.90%	28.80%
Not cancer	74.10%	71.20%

Best 2D model
80.0% accuracy (patient)
with threshold = 0.004

Best 3D model
73.5% accuracy (patient)



Hypothesis: nodules are small, rare, and indistinct; need better signal-to-noise ratio!

Follow-up Investigations

To improve our lung cancer prediction accuracy, we have identified another dataset (LUNA 2016) that contains more detailed annotations of lung nodules.

Work in progress:

- Prepare and augment training data for 3D image patches around nodules or other tissue
- Implement new model architecture for "sliding window" nodule classification