Cloud-Based Platform for COPDGene CT Images

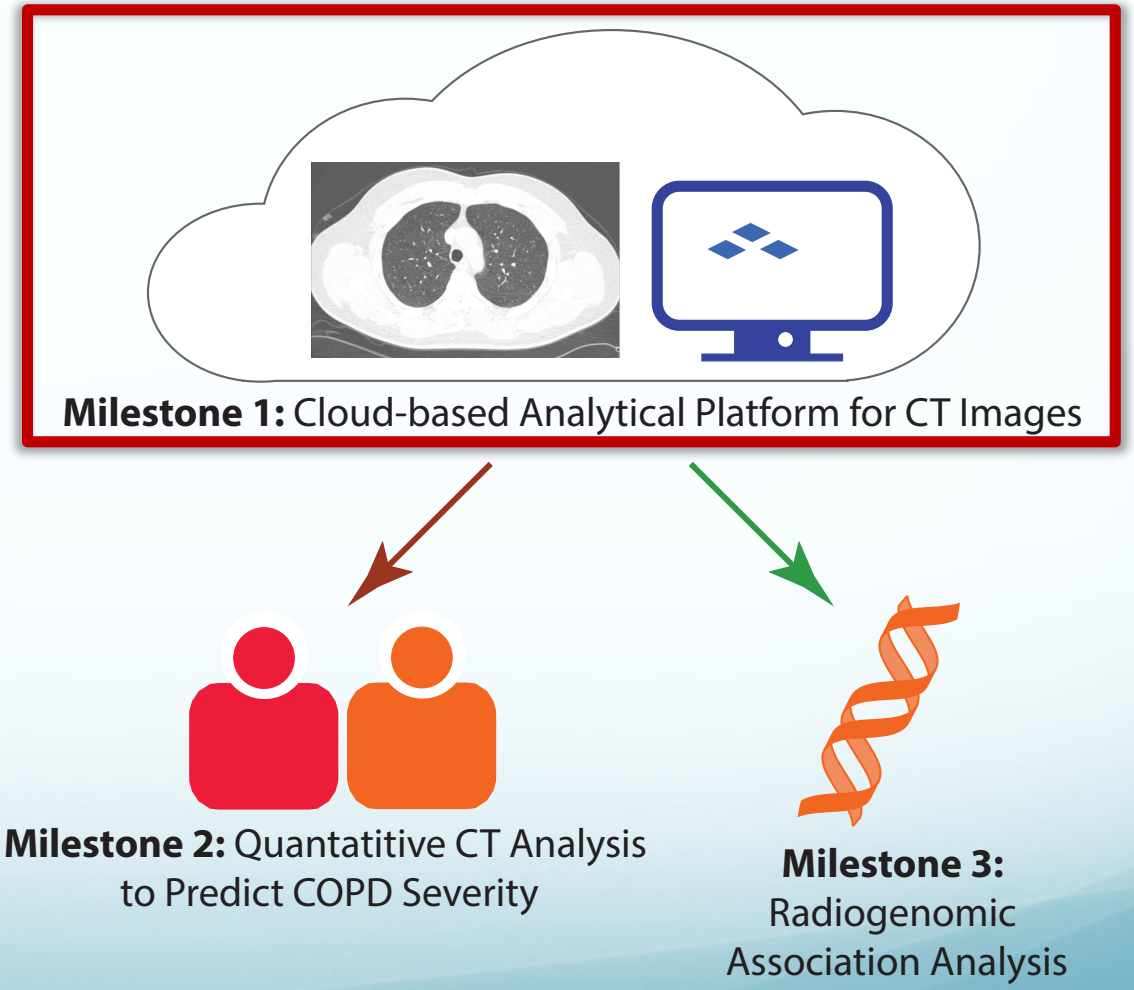
Kun-Hsing Yu, MD, PhD

Department of Biomedical Informatics
Harvard Medical School

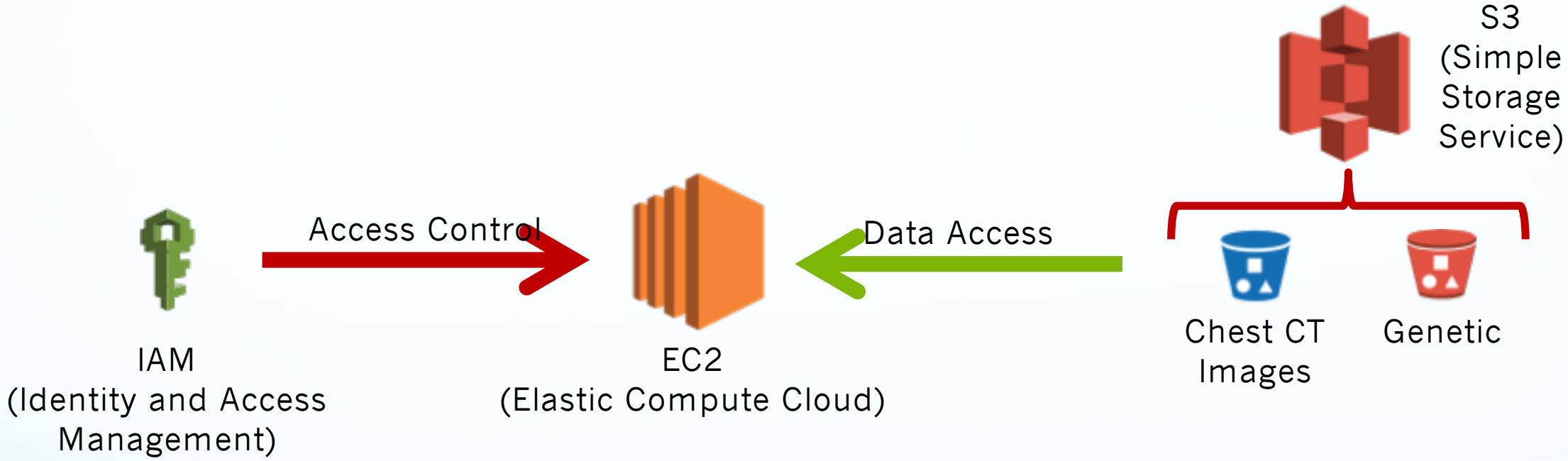
July 9th, 2018

Outline

- A secure cloud-based analytical platform for
  - Data management
  - Programming interface
  - Visualization
  - 2D and 3D convolutional neural networks analyses



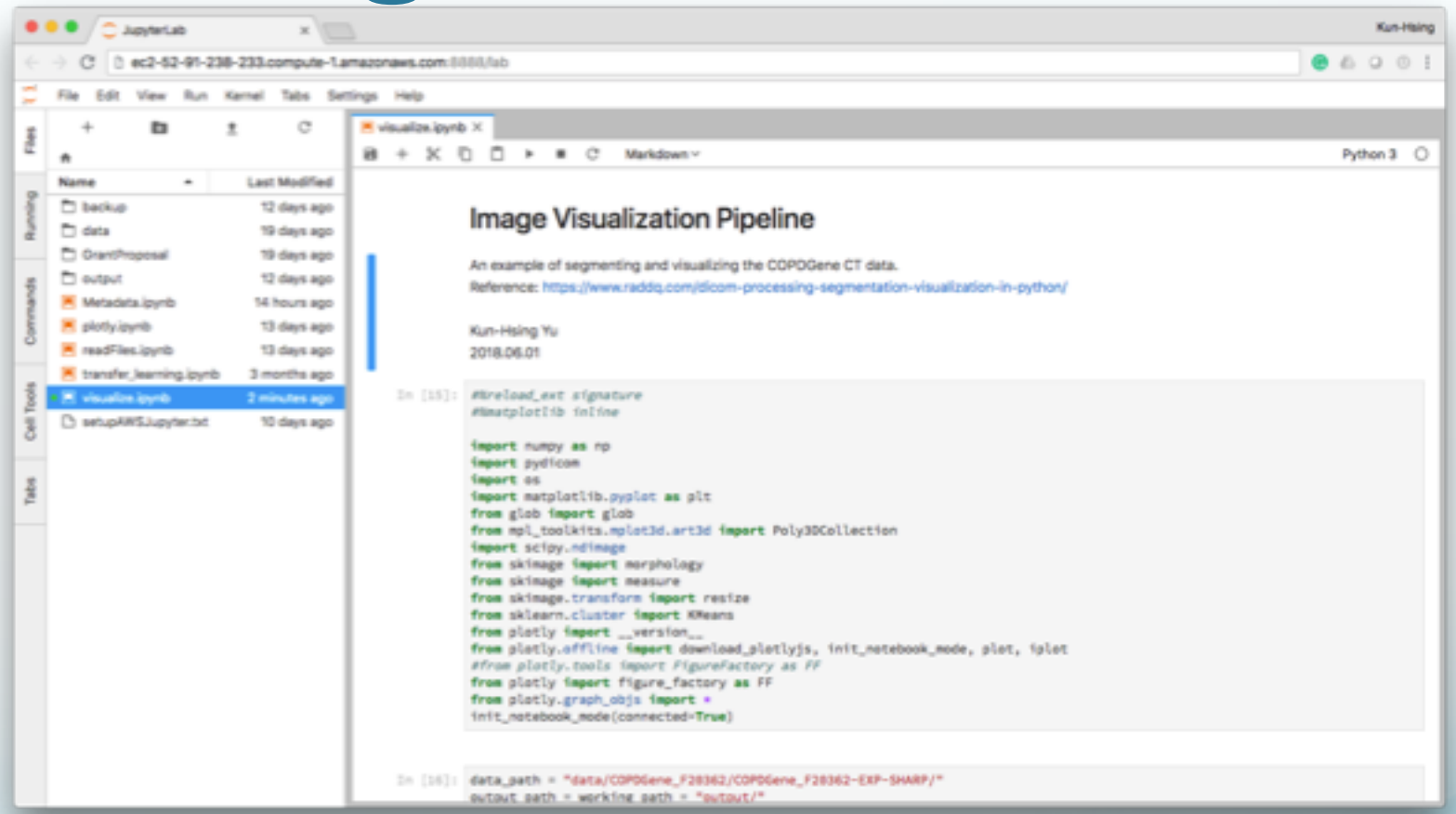
Platform



Analytical pipeline also validated on:

Programming Interface

- Scalable cloud computing
  - Tested on 4-GPU, 8-GPU, and 16-GPU virtual servers



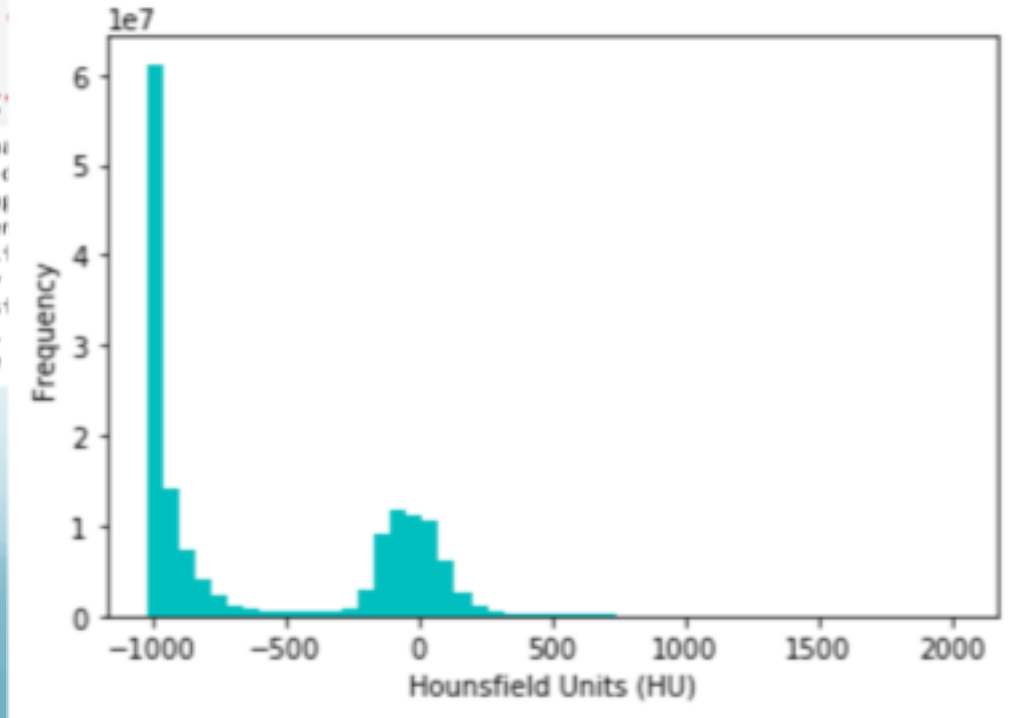
- Prototype: http://rebrand.ly/copdgene (password protected)

Data & Metadata Access

Metadata

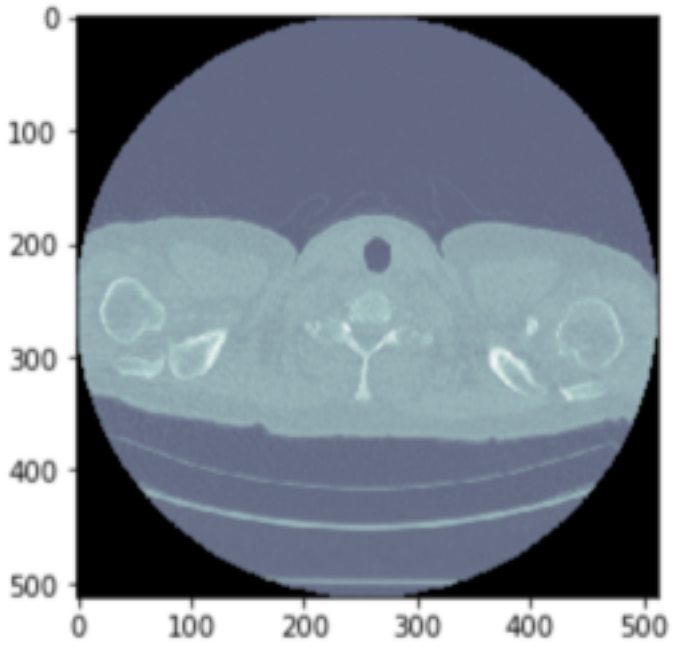
```
In [3]: print("Filename.........:", filename)
        print("Storage type.....:", dataset.SOPClassUID)
        print("Patient id.......:", dataset.PatientID)
        print("Modality.........:", dataset.Modality)
        print("Study Date.......:", dataset.StudyDate)
        if 'PixelData' in dataset:
            rows = int(dataset.Rows)
            cols = int(dataset.Columns)
            print ("Image size.......: {rows:d} x {cols:d}, {size:d} bytes".format(
            if
        print ('
```

```
('Filen
9.1732.
('Stora
('Patie
('Modal
('Study
Image si
('Pixel
('Slice
```

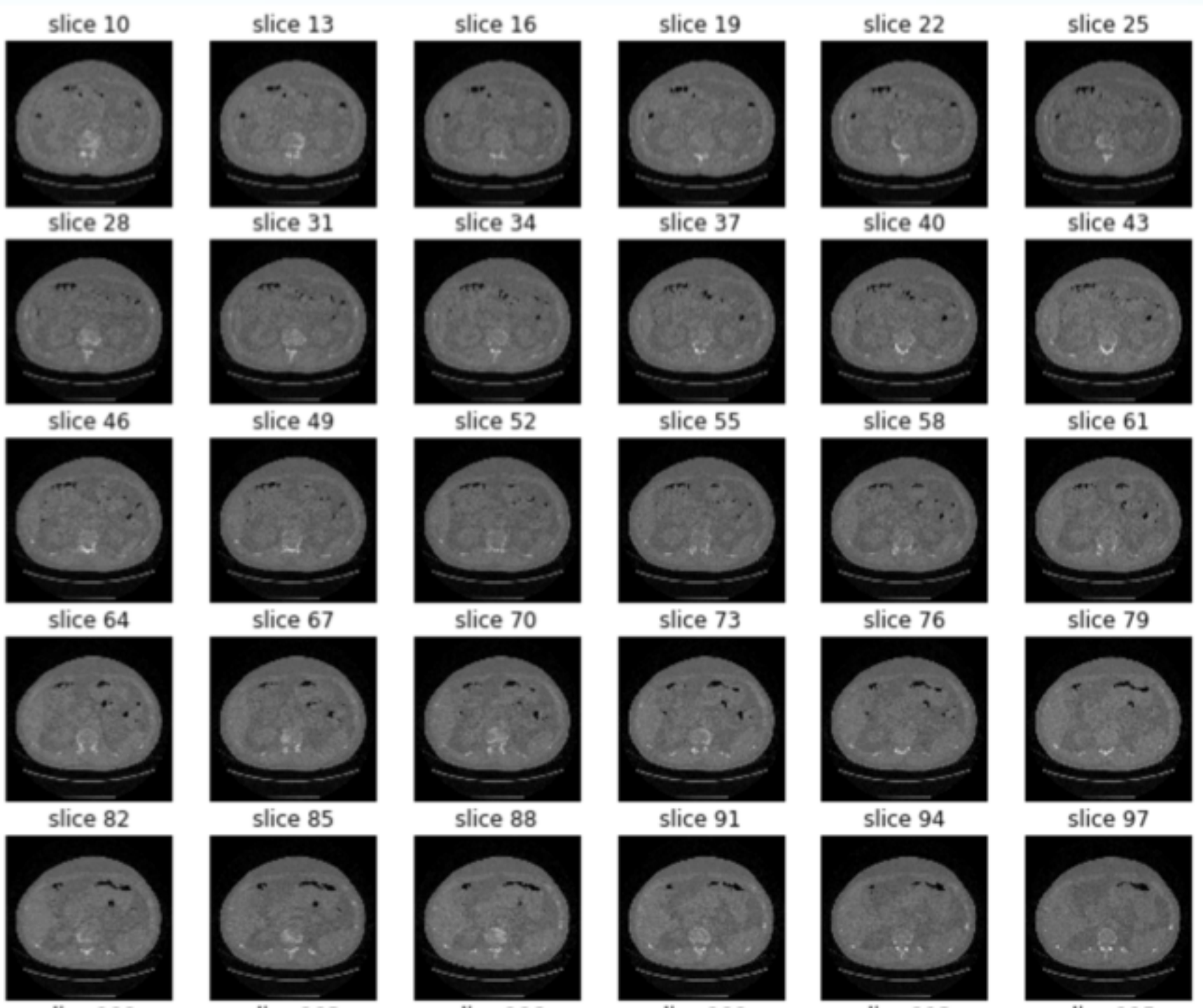


CT scan visualization and windowing

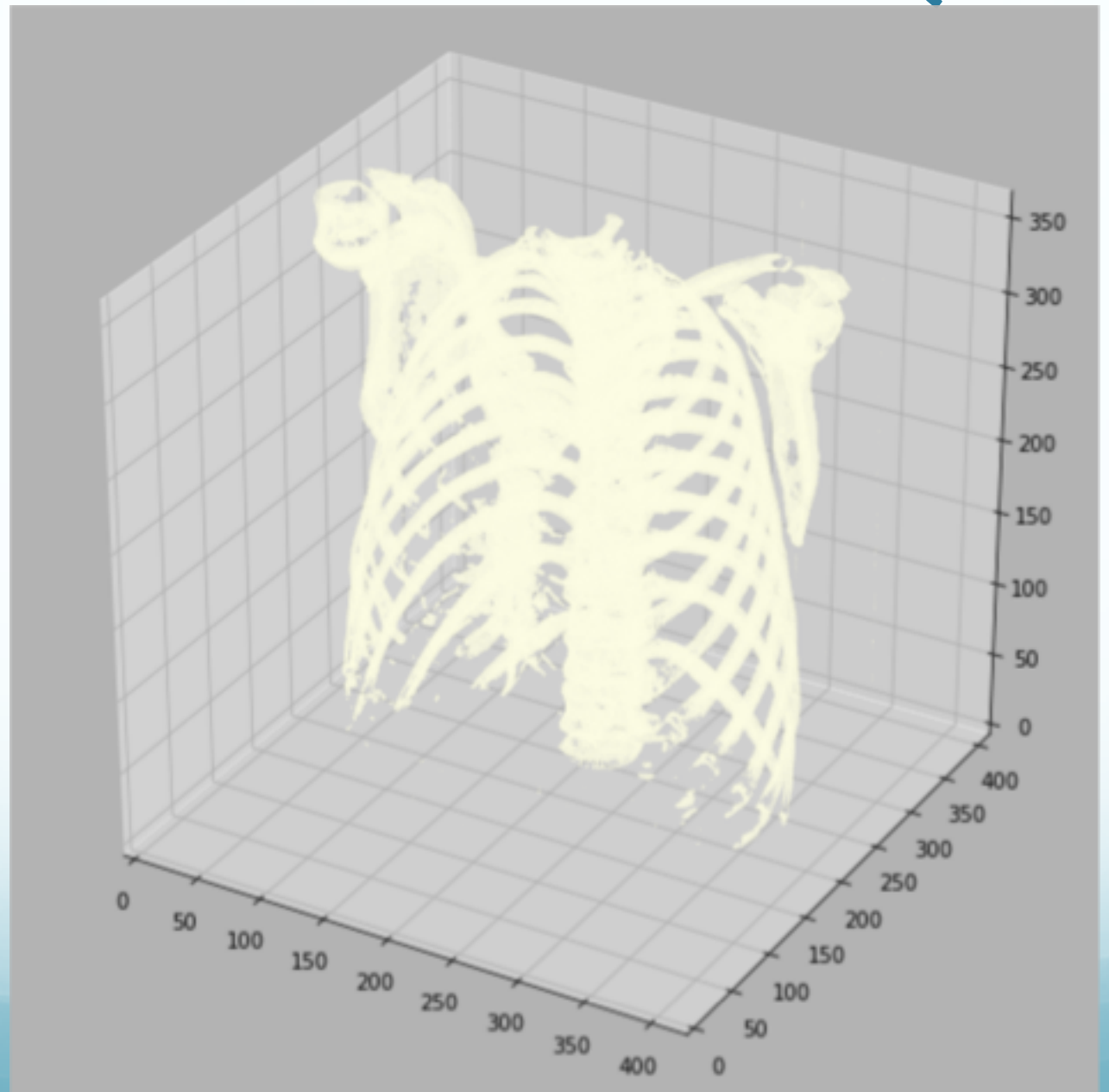
```
In [7]: plt.imshow(dataset.pixel_array, cmap=plt.cm.bone)
        plt.show()
```



CT Slice Visualization



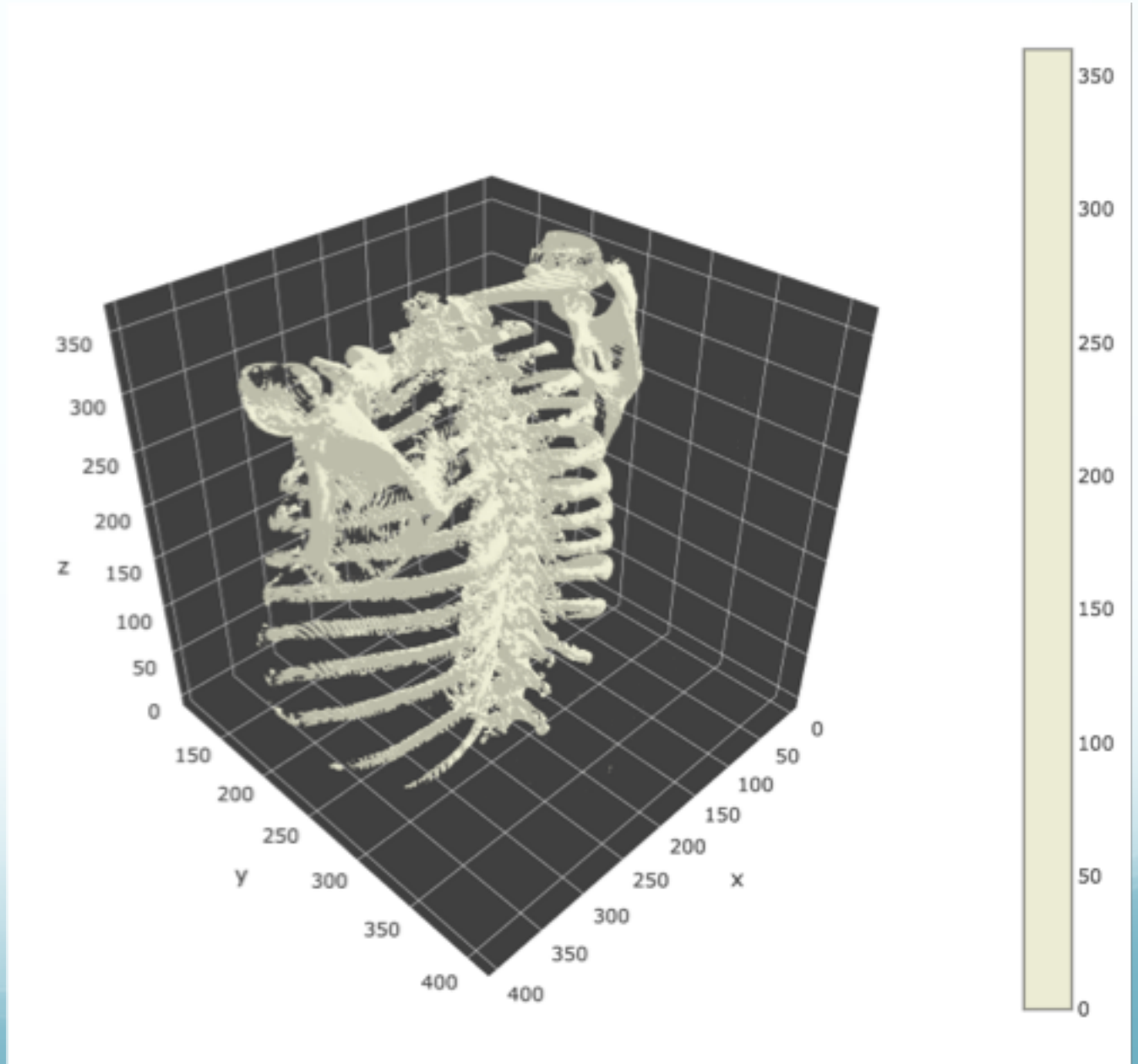
3D Reconstruction (1/2)



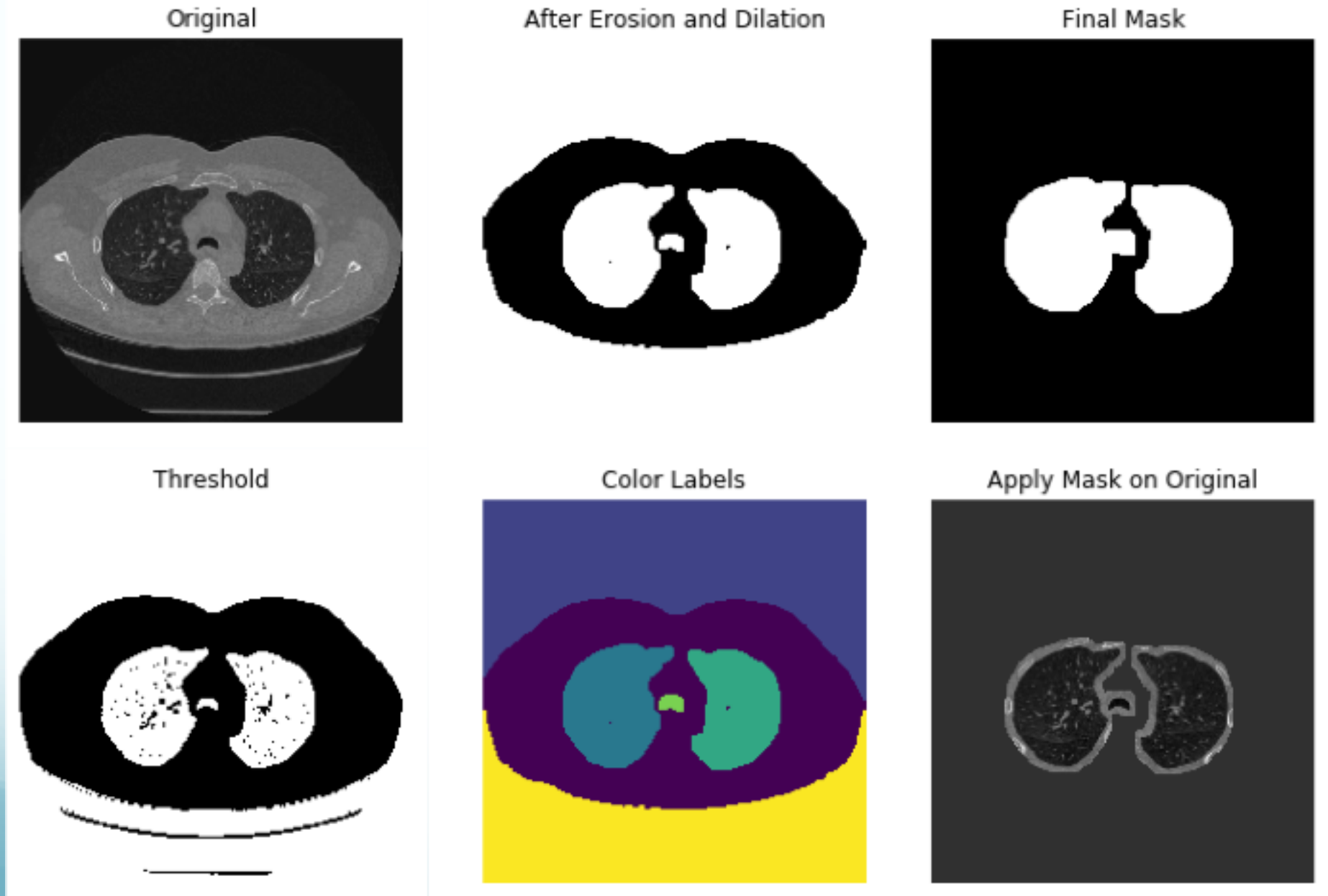
Computation time:
< 5s on a one-GPU AWS instance

3D Reconstruction (2/2)

- Interactive view
  - http://rebrand.ly/copdgene (password protected)

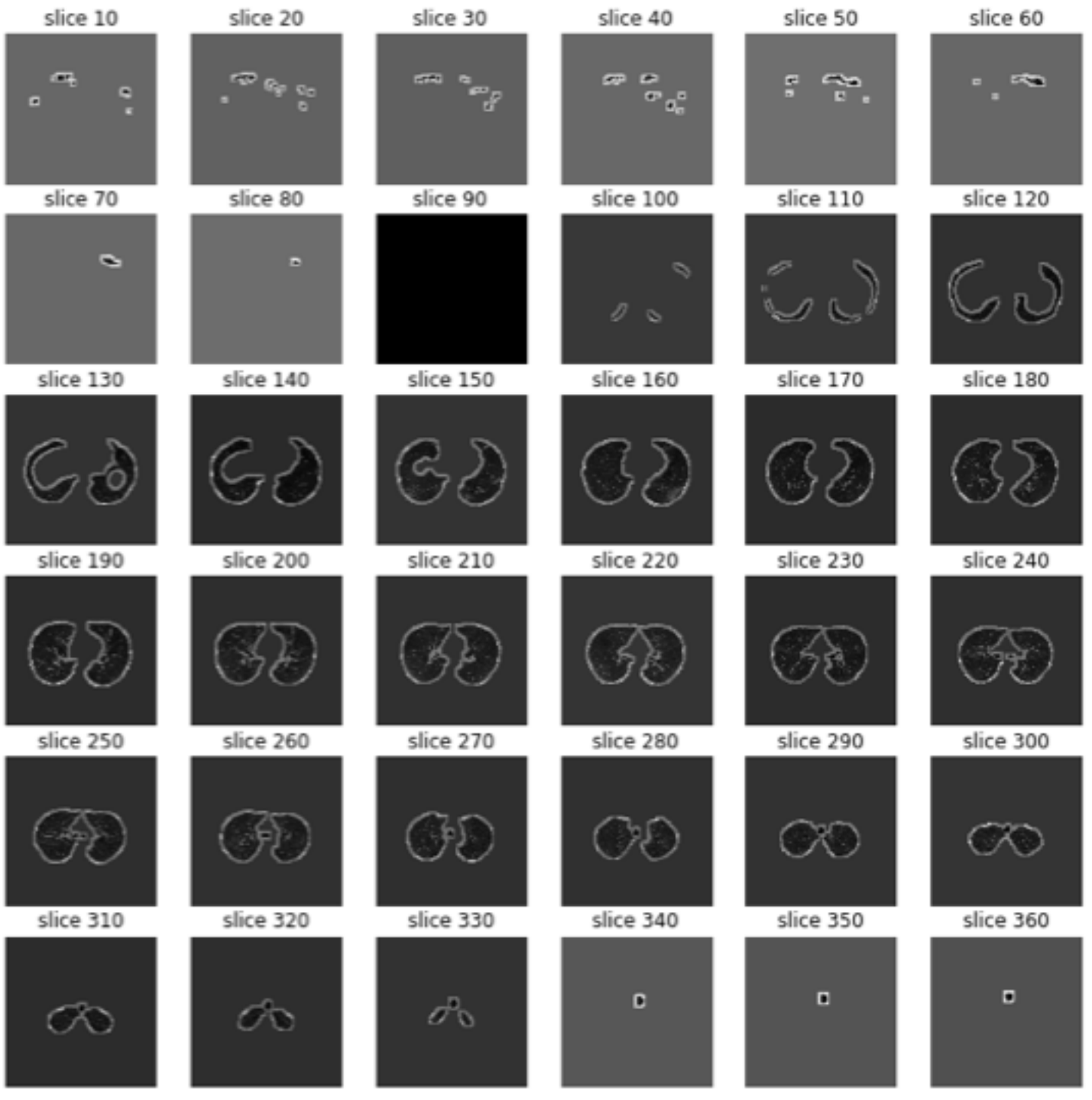


Cloud-Based Image Segmentation



Lung Segmentation

- Apply “lung masks” to all slices
  - Computation time: < 3s on a one-GPU AWS instance



Convolutional Neural Networks

- Supports Keras, Tensorflow, PyTorch, and Caffe

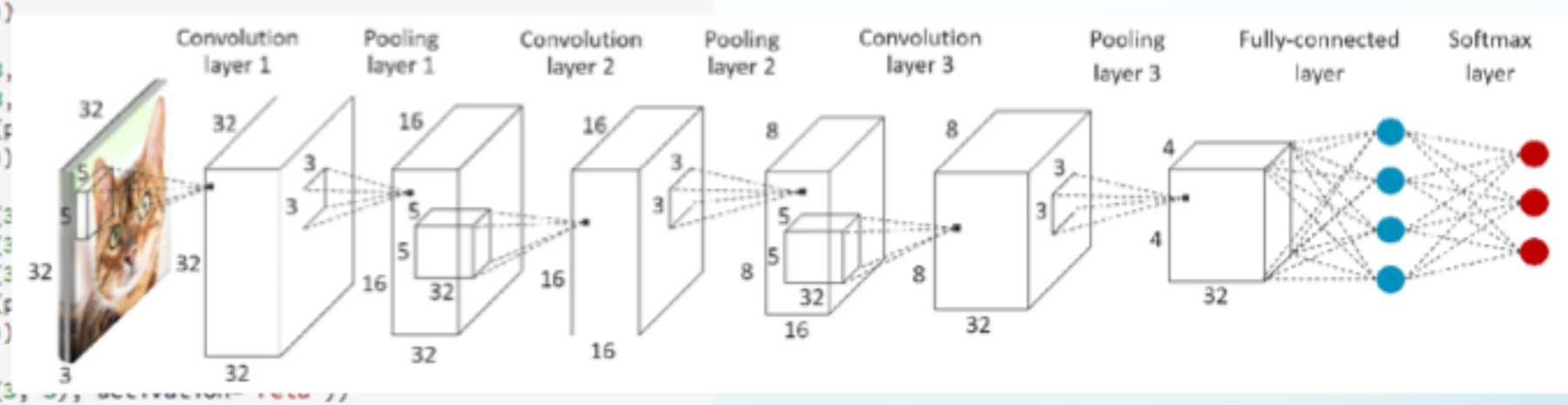
```
In [1]: from keras.models import Sequential
        from keras.layers import Dense, Dropout, Flatten
        from keras.layers import Conv2D, MaxPooling2D
        from keras.optimizers import SGD

        model = Sequential()

        model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 3)))
        model.add(Conv2D(32, (3, 3), activation='relu'))
        model.add(MaxPooling2D(pool_size=(2, 2)))
        model.add(Dropout(0.25))

        model.add(Conv2D(64, (3,
        model.add(Conv2D(64, (3,
        model.add(MaxPooling2D(p
        model.add(Dropout(0.25))

        model.add(Conv2D(128, (3
        model.add(Conv2D(128, (3
        model.add(Conv2D(128, (3
        model.add(MaxPooling2D(p
        model.add(Dropout(0.25))

        model.add(Conv2D(256, (3,
        model.add(Conv2D(256, (3, 3), activation='relu'))
        model.add(Conv2D(256, (3, 3), activation='relu'))
        model.add(MaxPooling2D(pool_size=(2, 2)))
        model.add(Dropout(0.25))

        model.add(Flatten())
        model.add(Dense(256, activation='relu'))
        model.add(Dropout(0.5))
        model.add(Dense(1, activation='sigmoid'))
```

TensorFlow

Keras

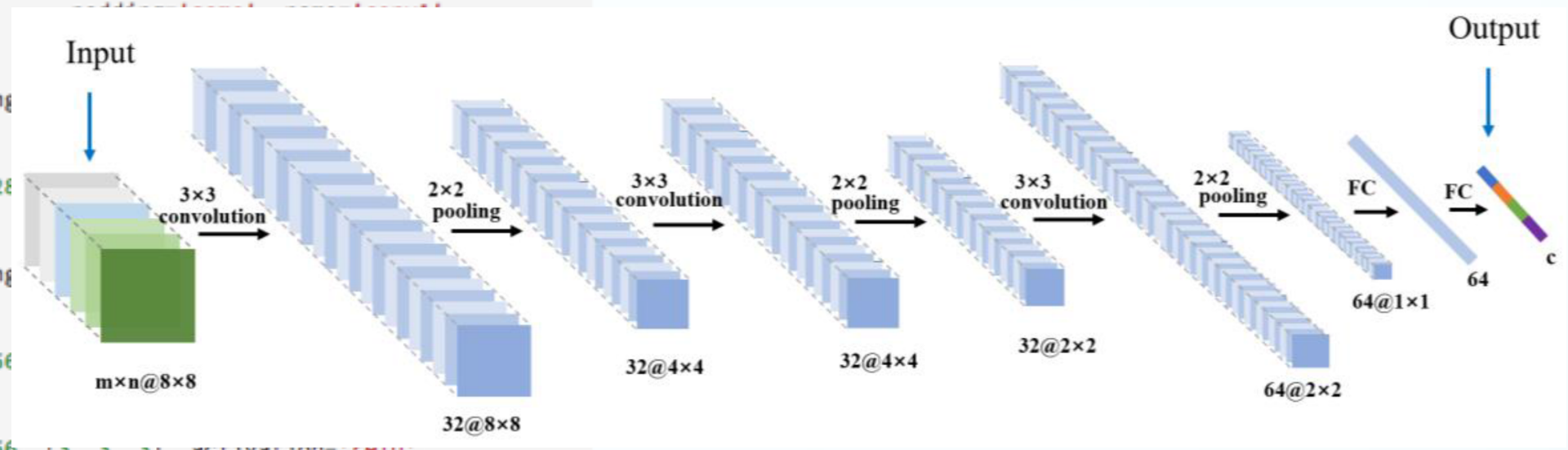
PyTorch

Caffe



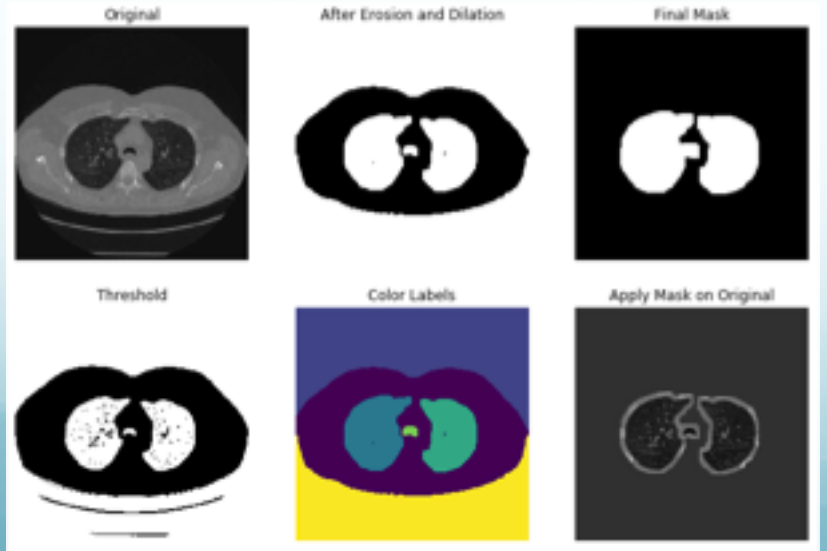
3D Convolutional Neural Networks

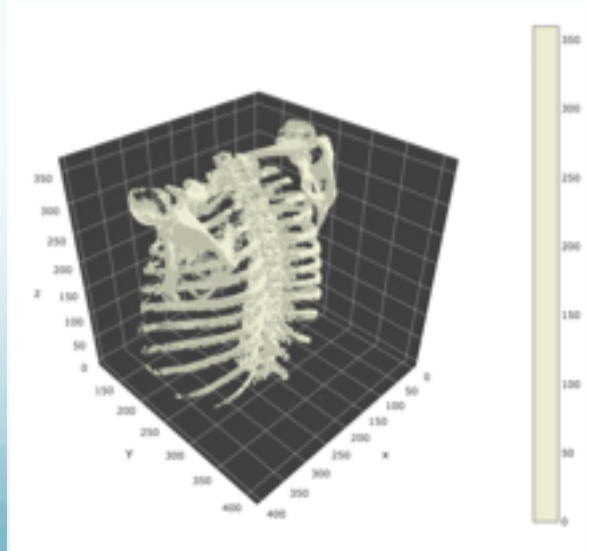
```
In [1]: model = Sequential()
        model.add(Conv3D(64, (3, 3, 3), activation='relu',
                         padding='same', name='conv1',
                         subsample=(1, 1, 1),
                         input_shape=(3, 16, 112, 112)))
        model.add(MaxPooling3D(pool_size=(1, 2, 2), strides=(1, 2, 2),
                               padding='valid', name='pool1'))
        # 2nd layer group
        model.add(Conv3D(128, (3, 3, 3), activation='relu',
                         padding='same', name='conv2',
                         subsample=(1, 1, 1)))
        model.add(MaxPooling3D(pool_size=(2, 2, 2), strides=(2, 2, 2),
                               padding='valid', name='pool2'))
        # 3rd layer group
        model.add(Conv3D(256, (3, 3, 3), activation='relu',
                         padding='same', name='conv3a',
                         subsample=(1, 1, 1)))
        model.add(Conv3D(256, (3, 3, 3), activation='relu',
                         padding='same', name='conv3b',
                         subsample=(1, 1, 1)))
        model.add(MaxPooling3D(pool_size=(2, 2, 2), strides=(2, 2, 2),
                               padding='valid', name='pool3'))
        # 4th layer group
        model.add(Conv3D(512, (3, 3, 3), activation='relu',
                         padding='same', name='conv4a',
                         subsample=(1, 1, 1)))
        model.add(Conv3D(512, (3, 3, 3), activation='relu',
                         padding='same', name='conv4b',
                         subsample=(1, 1, 1)))
        model.add(MaxPooling3D(pool_size=(2, 2, 2), strides=(2, 2, 2),
                               padding='valid', name='pool4'))
        # 5th layer group
        model.add(Conv3D(512, (3, 3, 3), activation='relu',
                         padding='same', name='conv5a',
                         subsample=(1, 1, 1)))
```



Summary

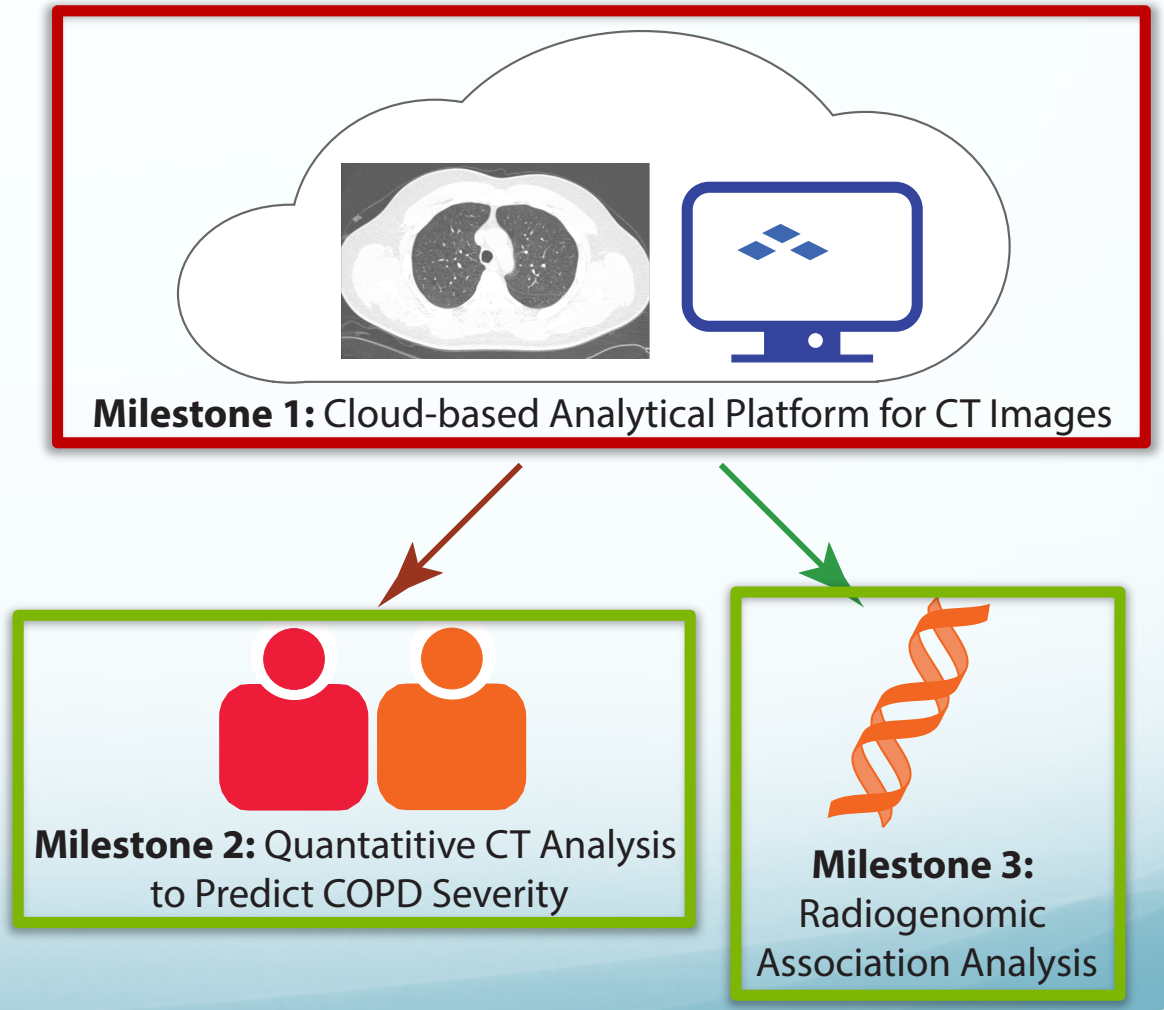
- Built a **cloud-based platform** for COPDGene CT images
  - Scalable computing through AWS
    - Data visualization, 3D reconstruction, lung segmentation
    - 2D and 3D convolutional neural networks
    - Deployed an Amazon Machine Image (AMI): **ami-7d6e4502**
  - Platform validated on Google Cloud and Microsoft Azure Cloud





Future Works

- To build convolutional neural network models using this framework
  - To predict COPD severity via quantitative CT analysis
  - To identify novel radio-genomic associations using the COPDgene cohort



Acknowledgements

- Zak Lab
  - Zak Kohane, MD, PhD
  - Arjun Manrai, PhD
- Avillach Lab
  - Paul Avillach, MD, PhD
  - Gregoire Versmee, MD
  - Laura Versmee, MD
- Harvard DBMI
  - Susanne Churchill, PhD
  - Chirag Patel, PhD
- Funding

Thank you. ☺

Kun-Hsing_Yu@hms.harvard.edu