

# Predicting Pneumonia

---

**Increasing the marketability  
of GE Healthcare's Critical  
Care Suite 2.01 (CCS)**





# Introduction



- ## Forecast Period 2022 - 2029
- 
- 

Market growth is expected to  
**ACCELERATE** at a CAGR of  
**5.7%**



---

One of the **KEY DRIVERS** for this  
market will be **RISING DEMAND**  
**FOR AI ASSISTED SYSTEMS**

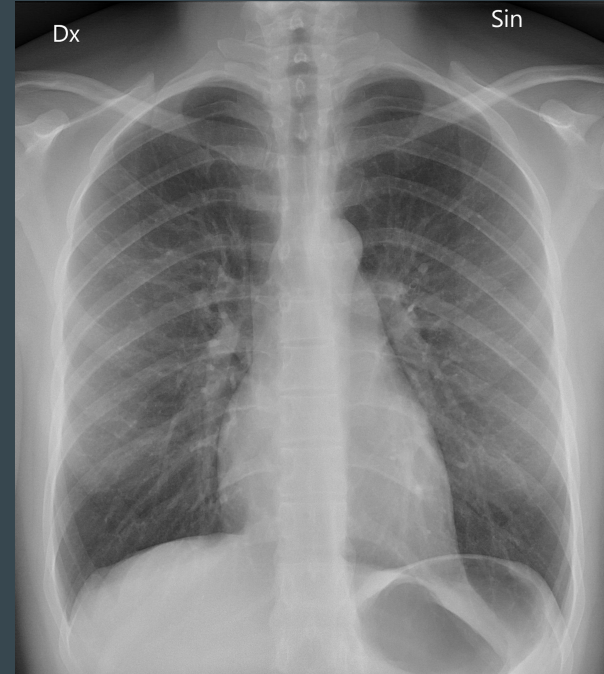




# Client Needs



# Increase the marketability of GE Healthcare's Critical Care Suite 2.01 (CCS)

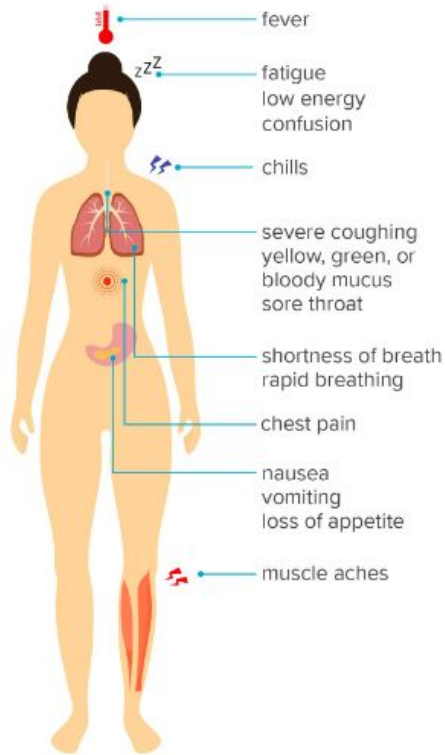




# Saving Lives



## Pneumonia



## Per Year



About 1 million adults in the US seek care in a hospital due to pneumonia



There are approximately 50,000 deaths in the U.S. alone.



Image from: Medical News Today, "Is it bronchitis or pneumonia?"



# The Goal







?



?

● ● ●

**Create a  
pneumonia  
recognition  
model.**

● ● ●



**Pneumonia**



**Normal**

• • •

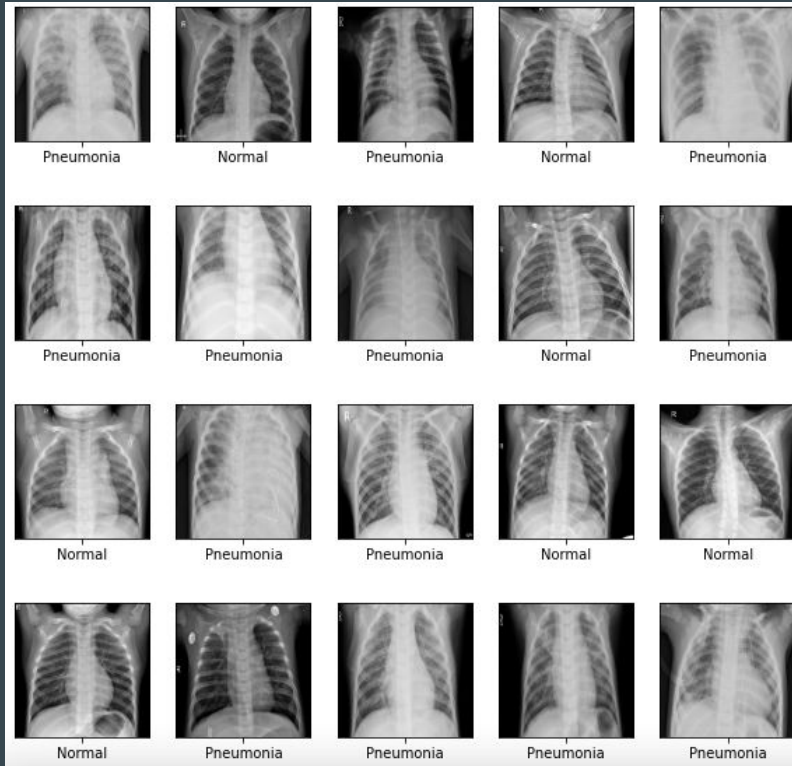
**Create a  
pneumonia  
recognition  
model.**

• • •



# Data Used





There are 5,856  
radiographs

- 1,583 NORMAL
- 4,273 PNEUMONIA  
(both bacterial and  
viral)

Data available on **Mendeley Data**



Preliminary models were trained with **500 images** (equal parts pneumonia and normal)

**Final model** was trained with **2,648 images** (equal parts pneumonia and normal)

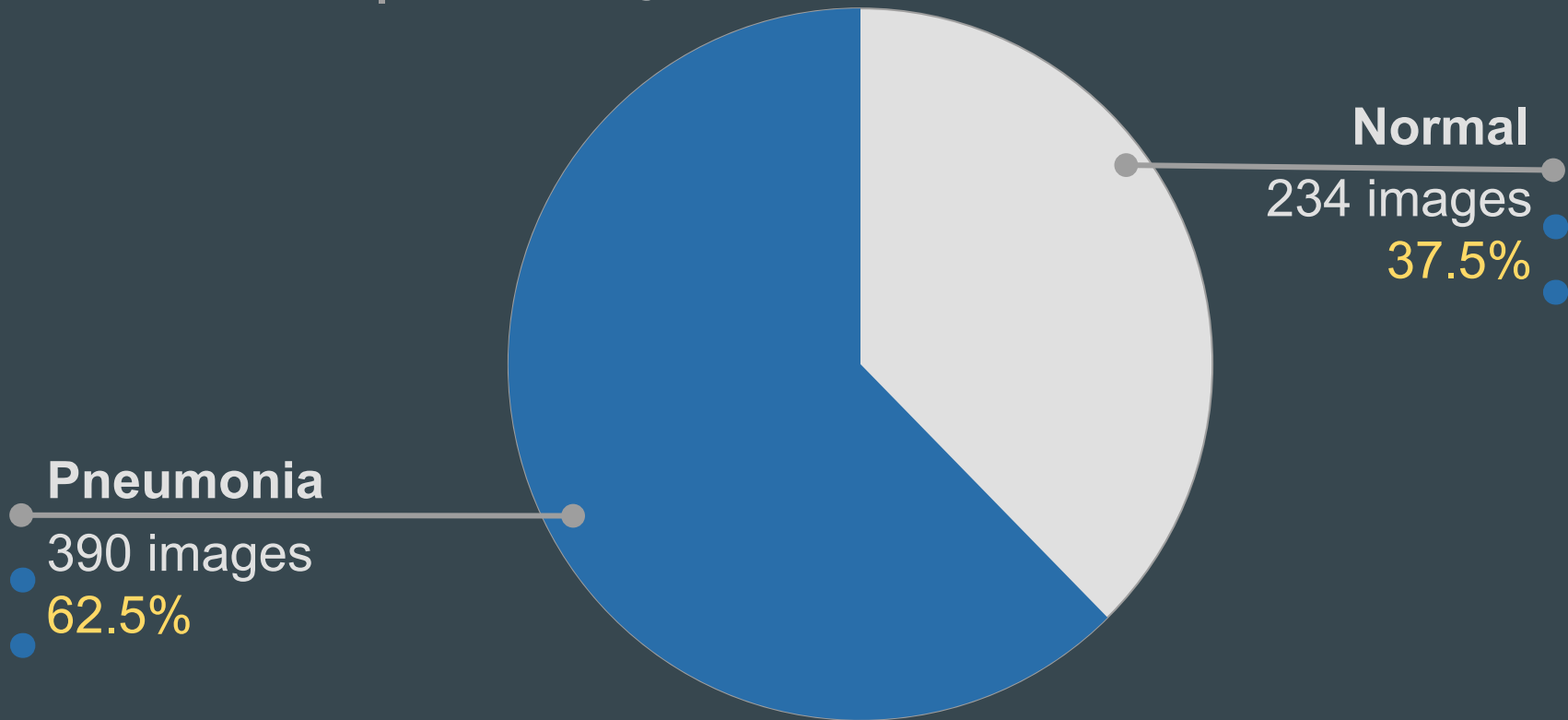


# Final Model

performance

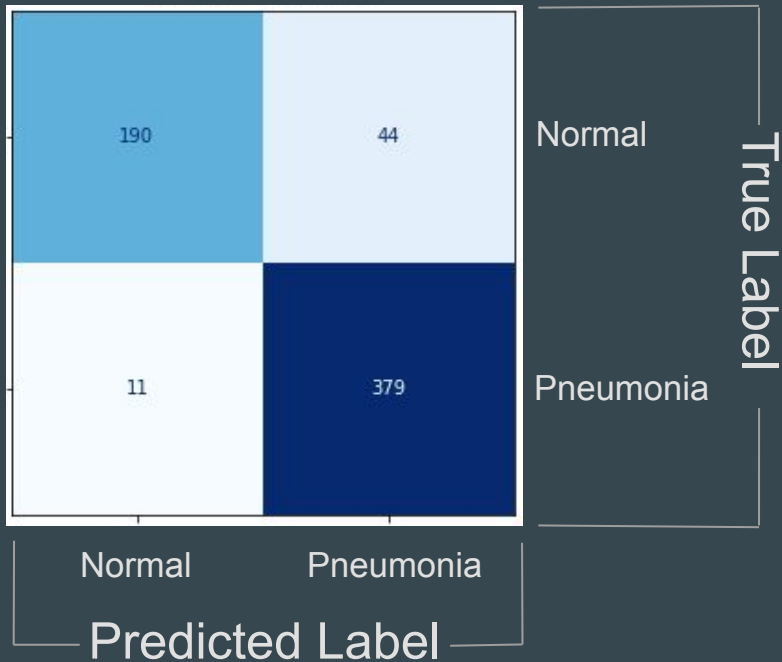


## Test Data | 624 images

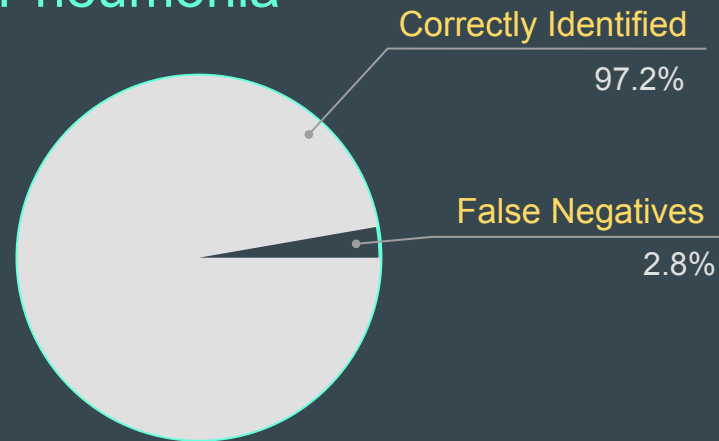


# Final Model

performance on unseen test data



## Pneumonia





# Final Model

---

performance on unseen test data

- ❏ Overall Accuracy **91.2%**
- ❏ Correctly Identified **97.2%**  
Pneumonia cases



# Next Steps





- ❏ Increase lambda coefficient in l2 regularization
- ❏ Reduce nodes further
- ❏ Remove a layer
- ❏ Decrease dropout p value
- ❏ Instead of downsampling Pneumonia,  
● upsample Normal  
●



# Wrap Up



Adding this model to CCS machines will:

---

- ❏ Increase the marketability of the CCS system
- ❏ Elevate patient experiences
- ❏ Improve clinical outcomes

...

# Thank You

...



...

# Data Used

in FINAL MODEL

...





## Data used to train Final Model:

- Training set
  - 1324 NORMAL
  - 1324 PNEUMONIA
- Validation set
  - 25 NORMAL
  - 25 PNEUMONIA



	pneumonia_recall	misclassified	accuracy
base_train	0.98	0.010	0.99
base_test	0.97	0.150	0.85
cnn_train	1.00	0.000	1.00
cnn_test	0.98	0.131	0.87
reg_train	1.00	0.000	1.00
reg_test	0.99	0.139	0.86
reduced_train	1.00	0.000	1.00
reduced_test	0.98	0.125	0.88
drop_train	1.00	0.002	1.00
drop_test	0.96	0.107	0.89
final_train	0.97	0.013	0.99
final_test	0.97	0.088	0.91

More data  
used to  
train the  
model

```
final_model = models.Sequential()
final_model.add(layers.Conv2D(50, (3, 3), padding='same', activation='relu', input_shape=(256, 256, 1)))
final_model.add(layers.MaxPooling2D((2, 2)))

final_model.add(layers.Conv2D(40, (3, 3), padding='same', activation='relu'))
final_model.add(layers.MaxPooling2D((2, 2)))

final_model.add(layers.Conv2D(35, (3, 3), padding='same', activation='relu'))
final_model.add(layers.MaxPooling2D((2, 2)))

final_model.add(layers.Conv2D(32, (3, 3), padding='same', activation='relu'))
final_model.add(layers.MaxPooling2D((2, 2)))

final_model.add(layers.Flatten())

final_model.add(layers.Dense(32, kernel_regularizer=regularizers.l2(.005), activation='relu'))
final_model.add(layers.Dense(16, kernel_regularizer=regularizers.l2(.005), activation='relu'))
final_model.add(layers.Dense(5, kernel_regularizer=regularizers.l2(.005), activation='relu'))
final_model.add(layers.Dense(1, activation='sigmoid'))

final_model.compile(optimizer='SGD',
                    loss='binary_crossentropy',
                    metrics=['accuracy'])

final_model.summary()
```

# What more data can do for a model

	train_loss	train_acc	test_loss	test_acc	loss_diff	acc_diff
<b>base_model</b>	0.040549	0.988000	0.598990	0.849359	0.558442	0.138641
<b>base_cnn</b>	0.001954	1.000000	0.899908	0.868590	0.897954	0.131410
<b>reg_cnn</b>	1.012247	1.000000	1.839380	0.862179	0.827133	0.137821
<b>reduced_nodes</b>	0.321644	1.000000	1.059986	0.875000	0.738342	0.125000
<b>dropout</b>	0.348353	0.998000	0.967050	0.892628	0.618697	0.105372
<b>final_model</b>	0.113301	0.986782	0.449658	0.911859	0.336356	0.074924










Train

Test

Differences  
between train  
and test

< > CellData





Name

- ✓  chest\_xray
  - ✓  test
    - >  NORMAL
    - >  PNEUMONIA
  - ✓  train
    - >  NORMAL
    - >  PNEUMONIA
- >  code
- >  OCT

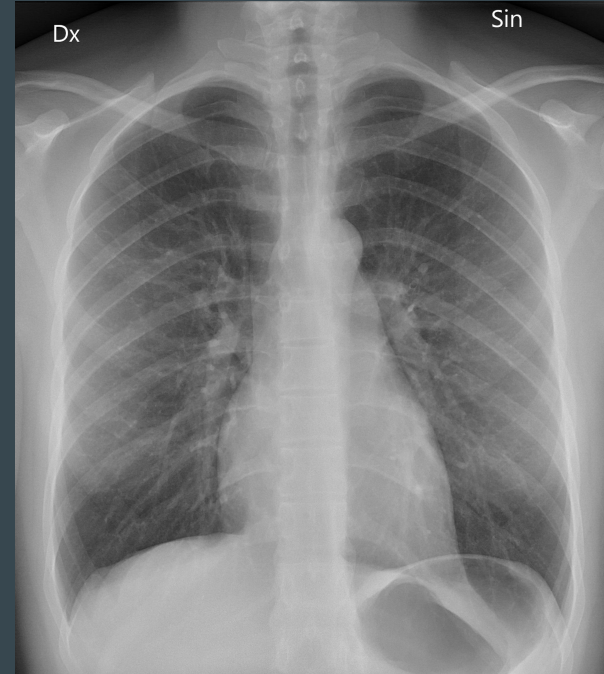
...

# Recommendations

...

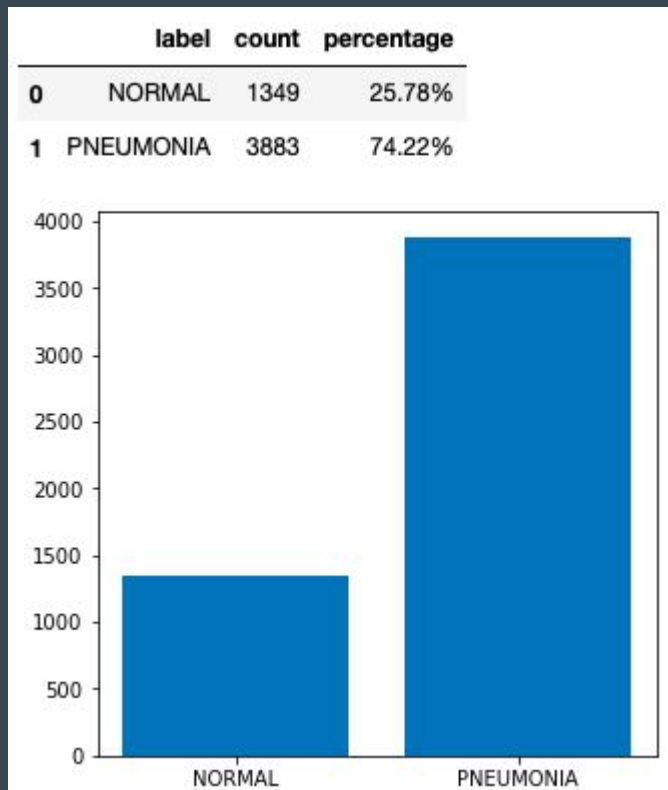
- 
- 
- Include a pneumonia recognition model on CCS systems
  - Radiologist confirm or reject predictions
  - Push software updates
- 
- 

Increase prospective customer base by increasing the application potential of CCS system.





## Distribution of all the training data



Preliminary models  
downsampled to:

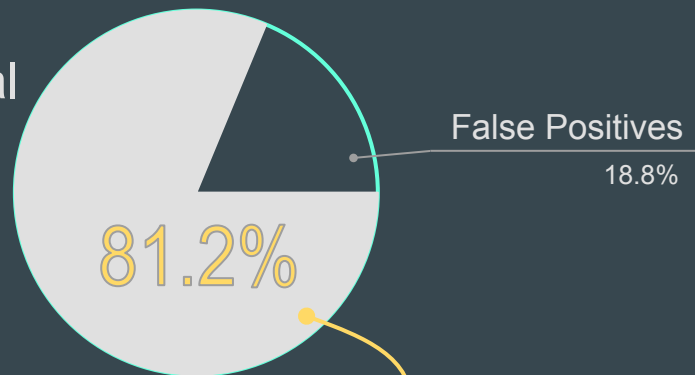
- Training set
  - 250 NORMAL
  - 250 PNEUMONIA
- Validation set
  - 25 NORMAL
  - 25 PNEUMONIA

# Final Model

performance on unseen test data

		True Label	
Predicted Label	Normal	Pneumonia	
	190	44	Normal
11	379		Pneumonia

Normal



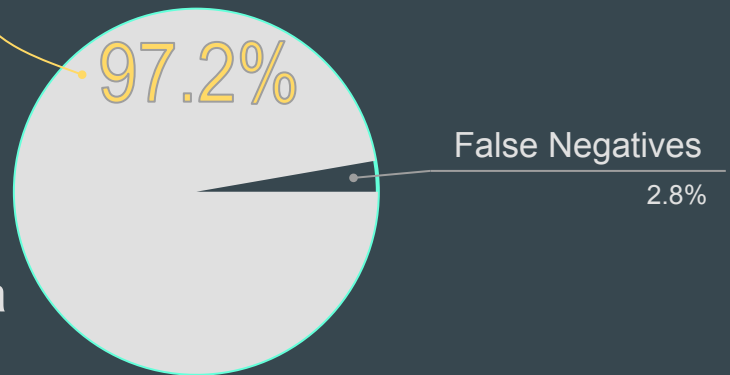
Correctly Identified

81.2%

False Negatives

2.8%

Pneumonia





● ● ●

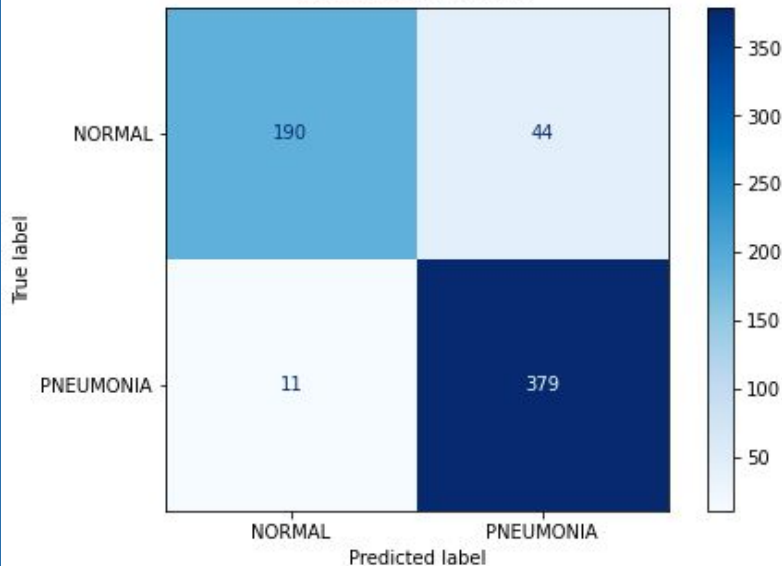
# Create a pneumonia recognition model.

● ● ●

# Final Model performance on unseen test data:

This model mislabeled 55 images (8.81%)

Confusion Matrix



Classification Report

