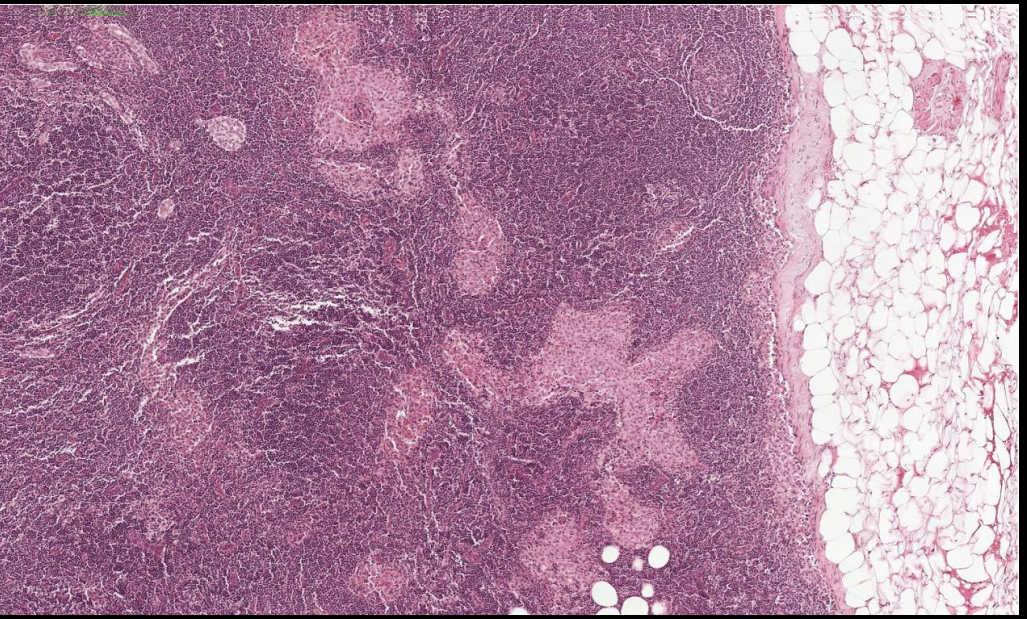
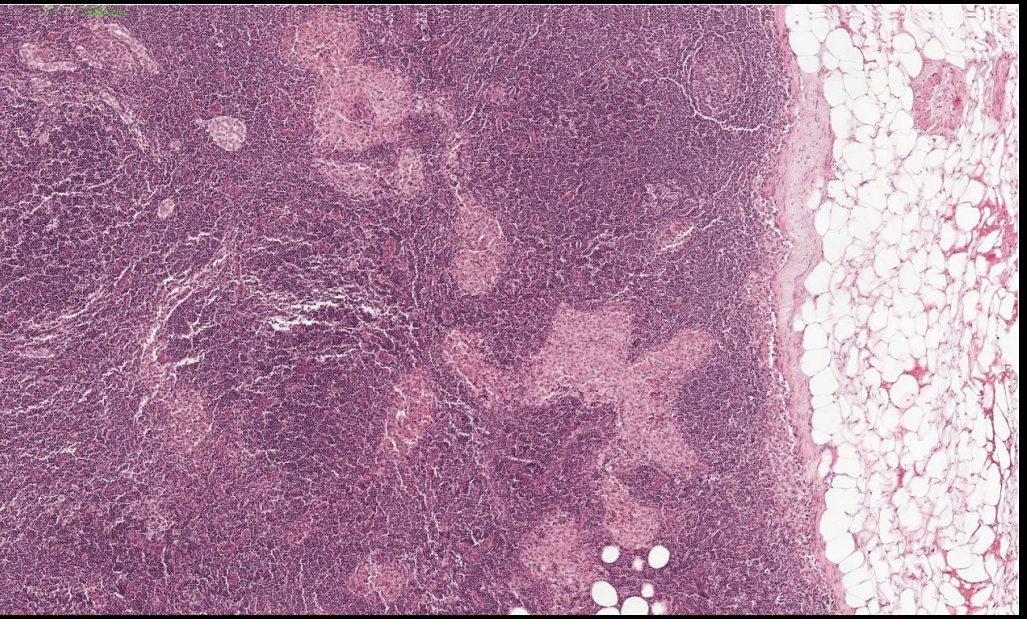
Medical Anomaly Edition

ft. GANs

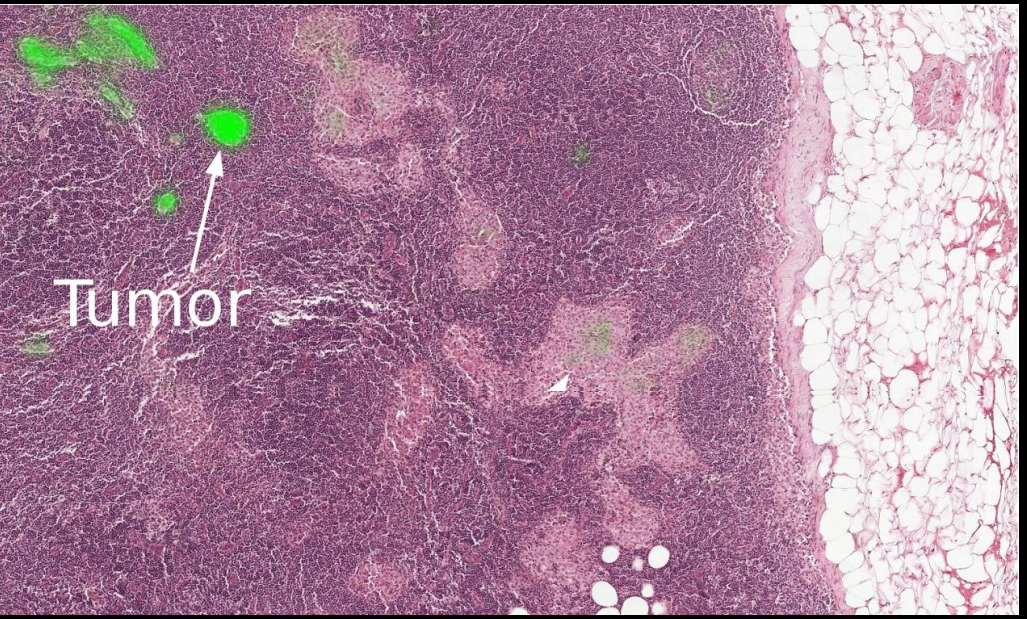

DR. Obot




DR. Obot



- Blastoma
- Cystic fibrosis
- Appendicitis
- Pneumonia
- Tumor



- ✗ Blastoma
- ✗ Cystic fibrosis
- ✗ Appendicitis
- ✗ Pneumonia
- ✓ Tumor

- Blastoma
- Cystic fibrosis
- Appendicitis
- Pneumonia
- Tumor

- Blastoma
- Cystic fibrosis
- Appendicitis
- Pneumonia
- Tumor

Alzheimer's

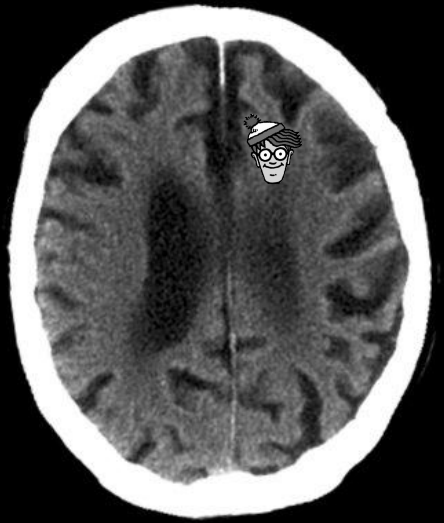
- ❌ Blastoma
- ❌ Cystic fibrosis
- ❌ Appendicitis
- ❌ Pneumonia
- ❌ Tumor

Alzheimer's

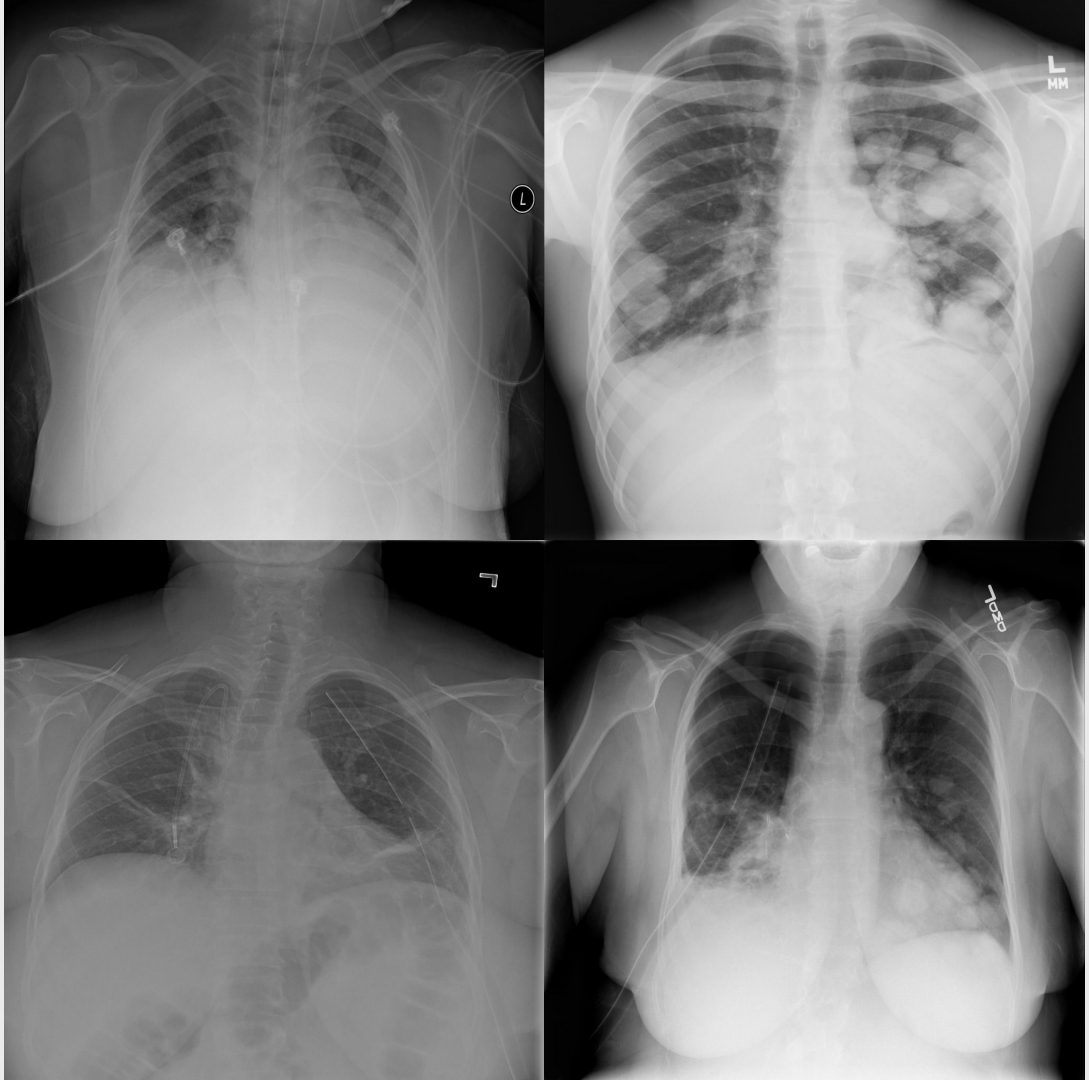
- ✗ Blastoma
- ✗ Cystic fibrosis
- ✗ Appendicitis
- ✗ Pneumonia
- ✗ Tumor

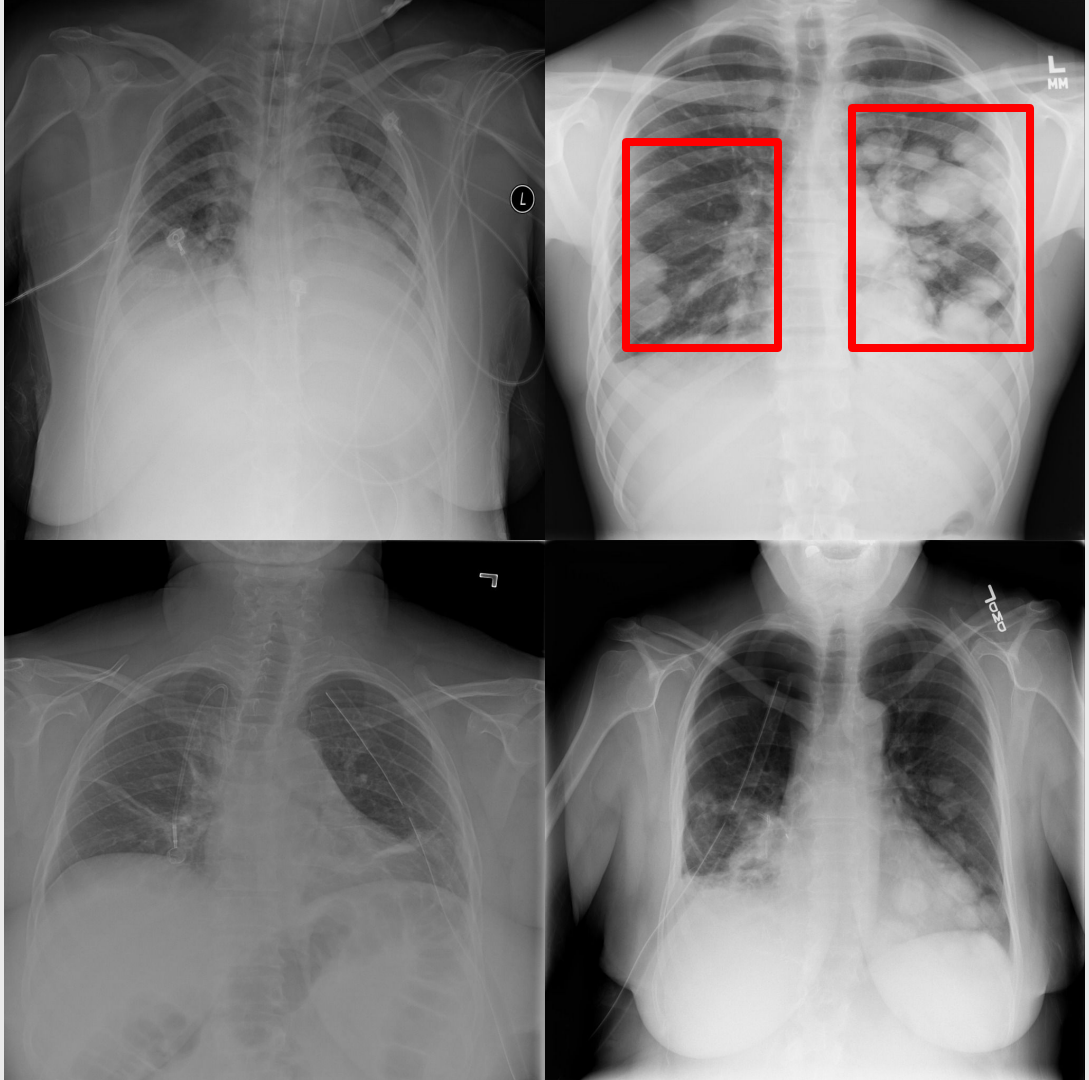
- ? Alzheimer's

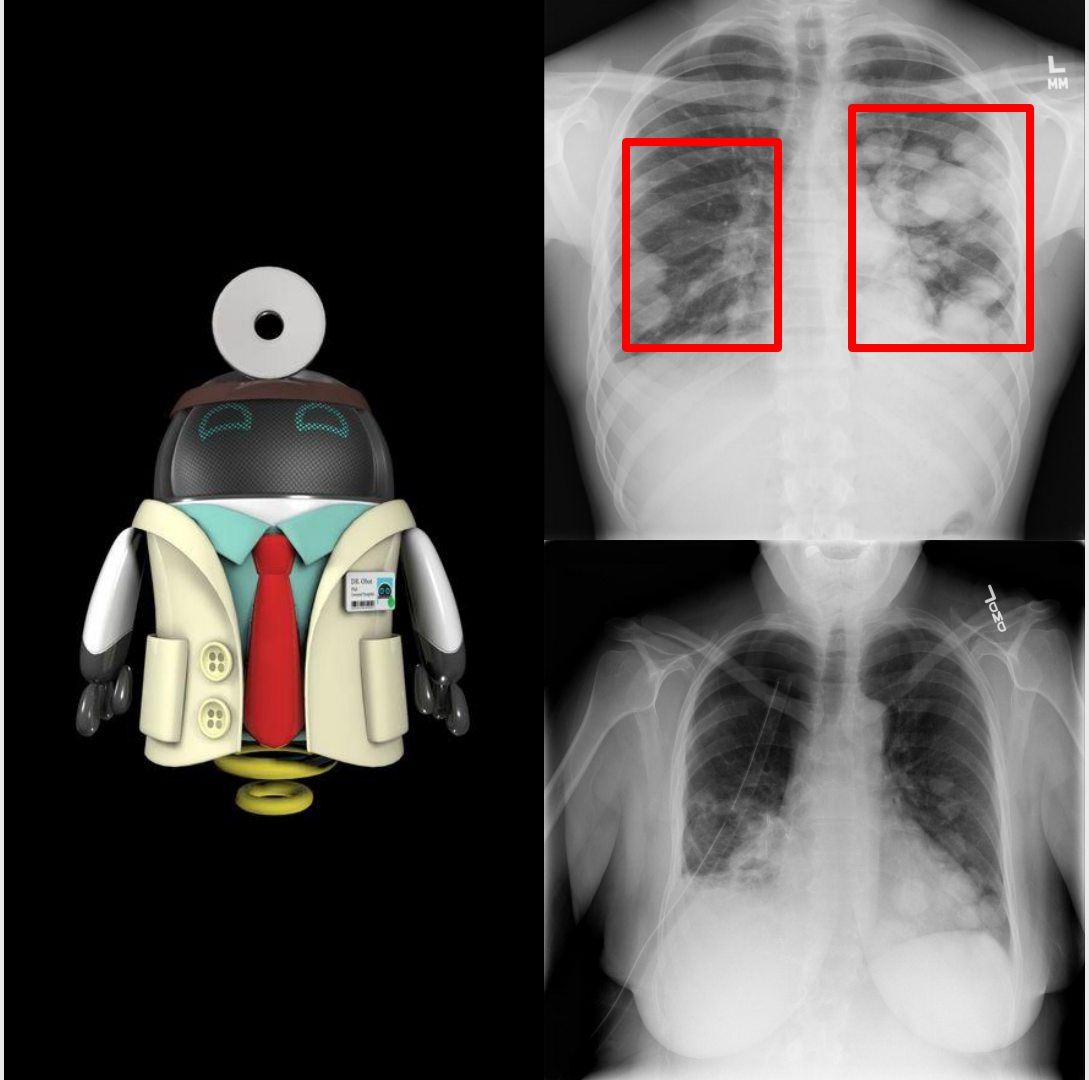
You only find what you’re looking for



- Blastoma
- Cystic fibrosis
- Appendicitis
- Pneumonia
- Tumor
- Alzheimer’s







ZILC

D: Detective

R: Real Data

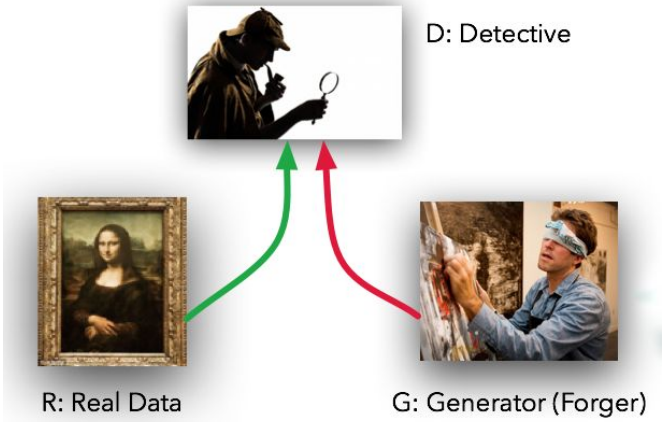
G: Generator (Forger)

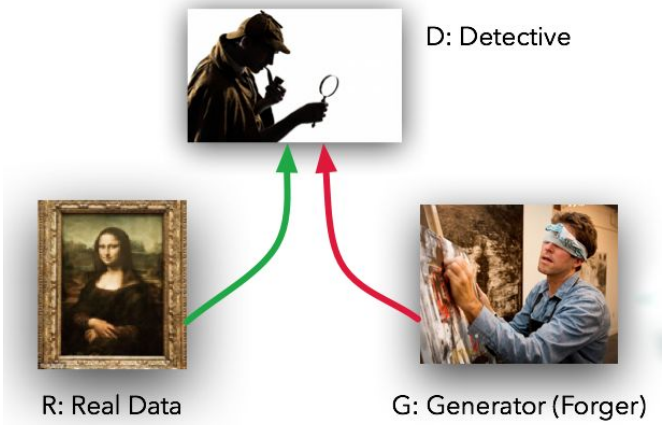
```
[[[232 234 234]
  [232 236 237]
  [228 231 235]
  ...
  [221 226 227]
  [214 224 224]
  [209 222 220]]

 [[231 233 233]
  [229 233 234]
  [227 231 236]
  ...
  [217 224 227]
  [216 226 226]
  [211 222 220]]
```

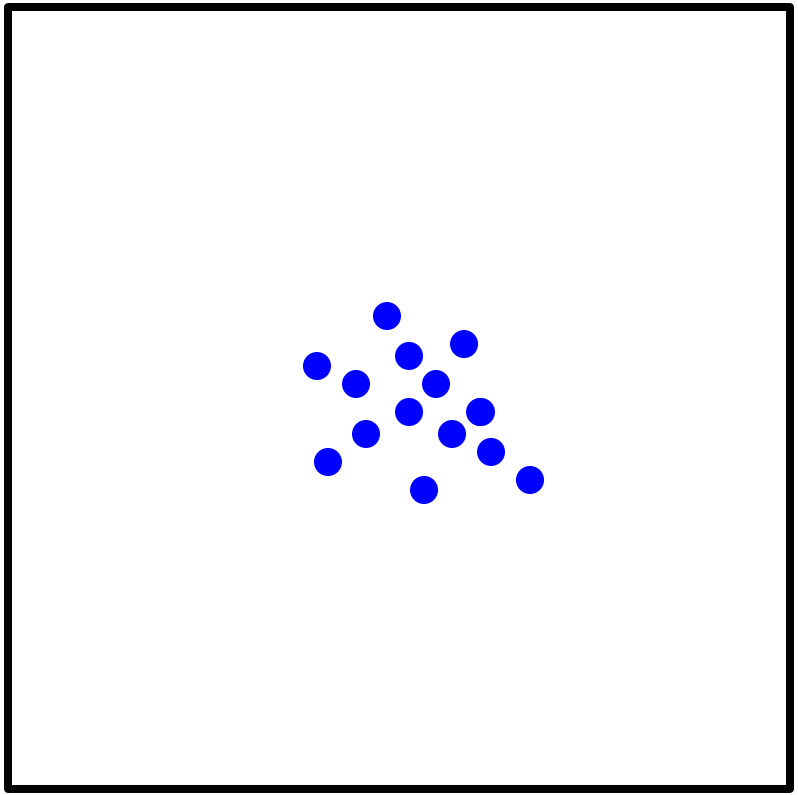
Color:	**Pink**
Shape:	**Circle**
Radius:	**12cm**
Icing?:	**Yes**
Hole?:	**Yes**
Hole Size:	**2.5cm**
Sprinkles?:	**Yes**
Sprinkles Size:	**1cm x 0.2cm**

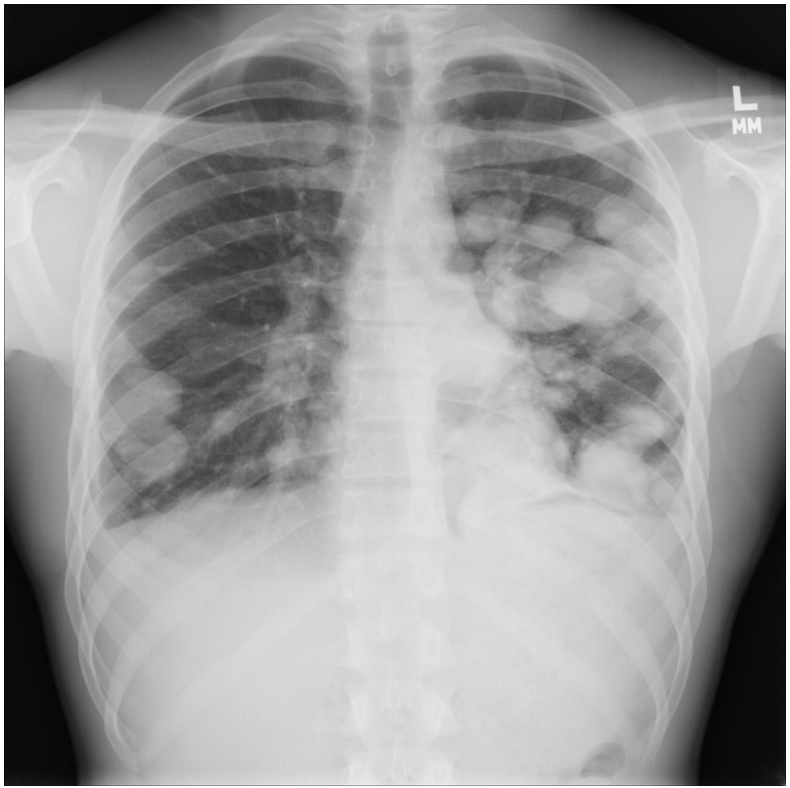
Color:	**(200, 130, 150)**
Shape:	**0**
Radius:	**12cm**
Icing?:	**1**
Hole?:	**1**
Hole Size:	**2.5cm**
Sprinkles?:	**1**
Sprinkles Size:	**1cm x 0.2cm**

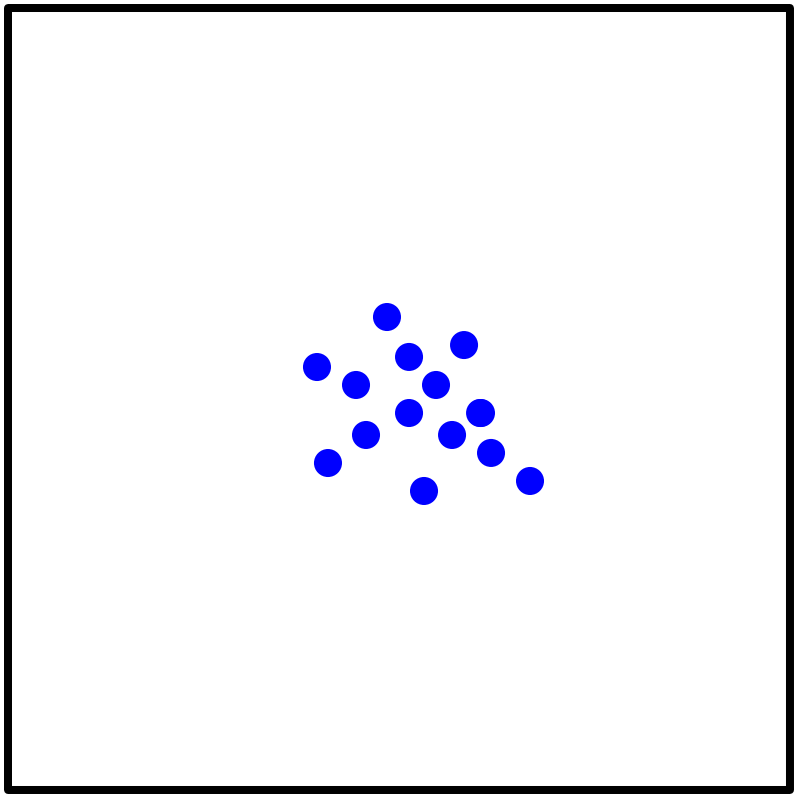
20
78
96
...
83
92
48
63


D: Detective
R: Real Data
G: Generator (Forger)

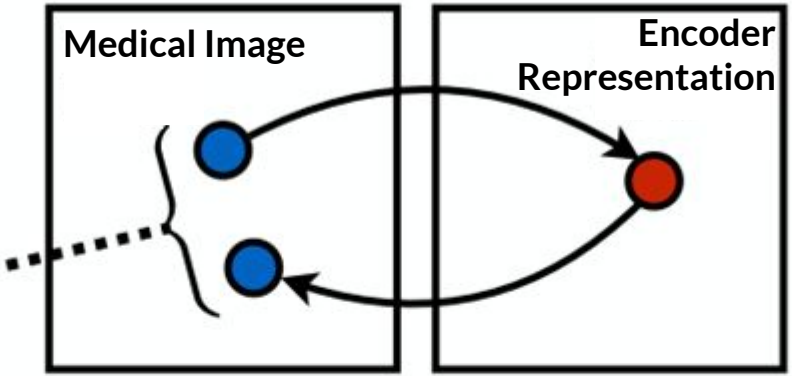

D: Detective
R: Real Data
G: Generator (Forger)




L
MM



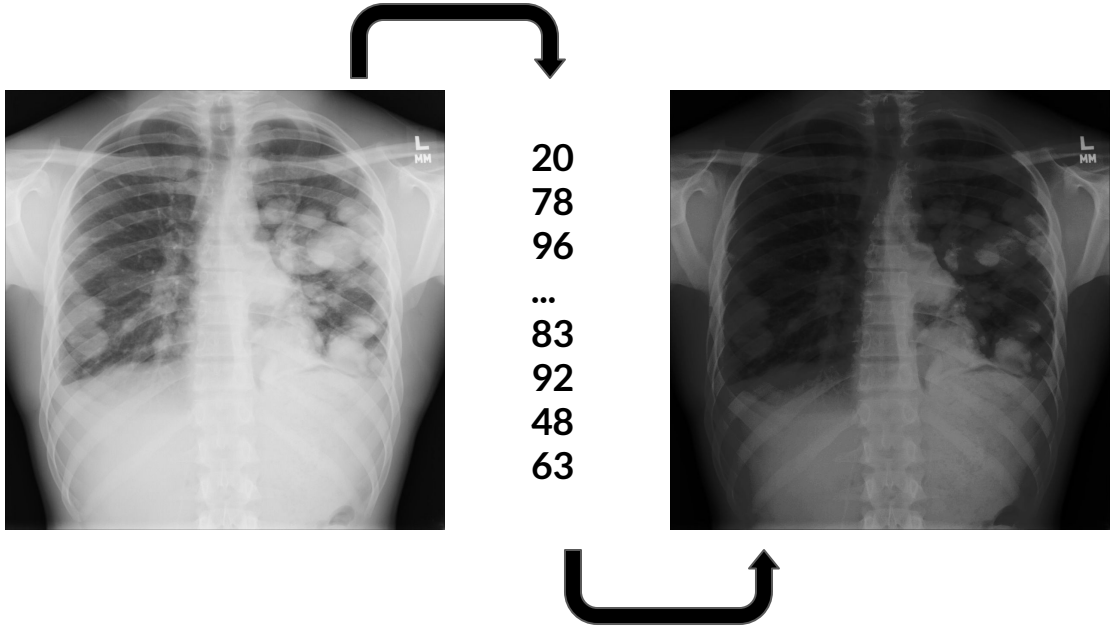
cycle-consistency loss



Reconstruction Loss

Encoder Matching

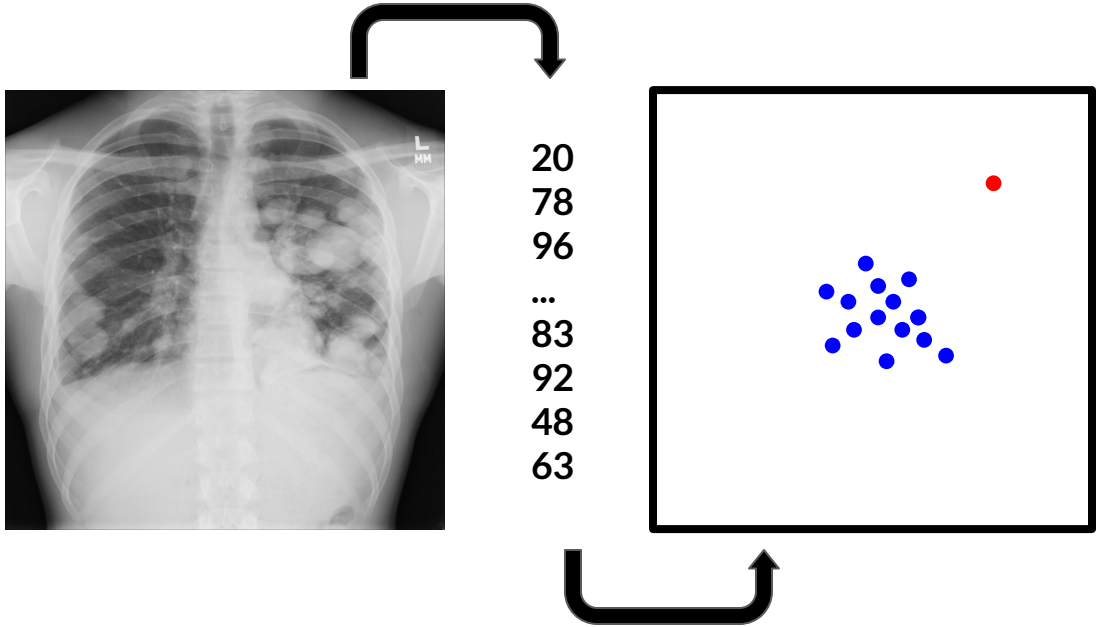
Discrimination



Reconstruction Loss

Encoder Matching

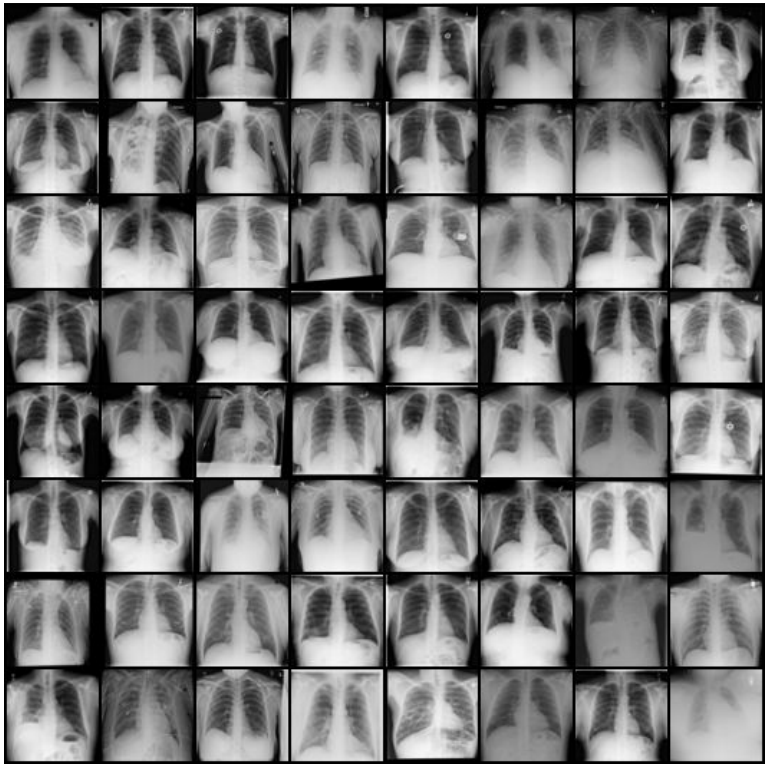
Discrimination

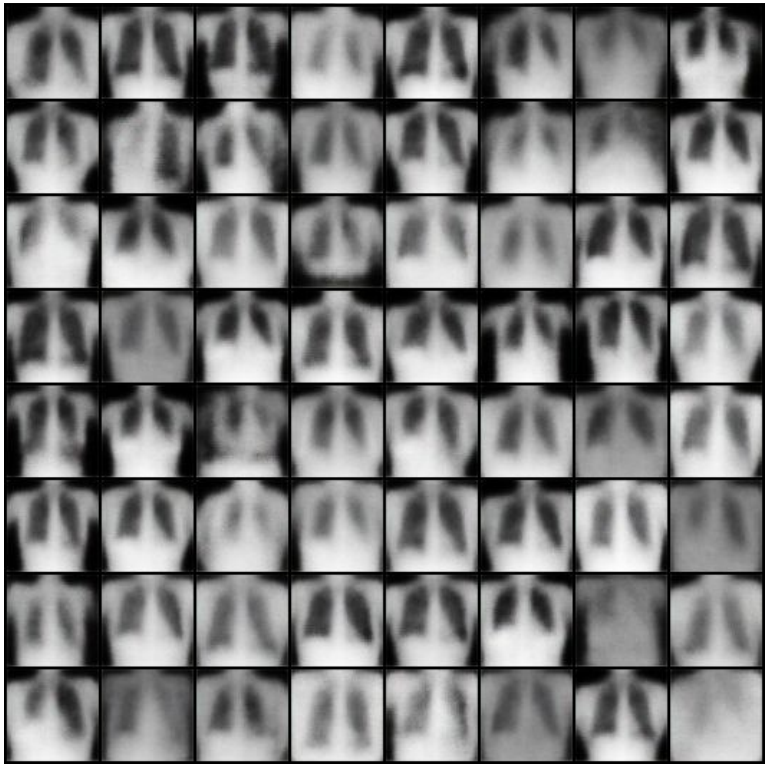


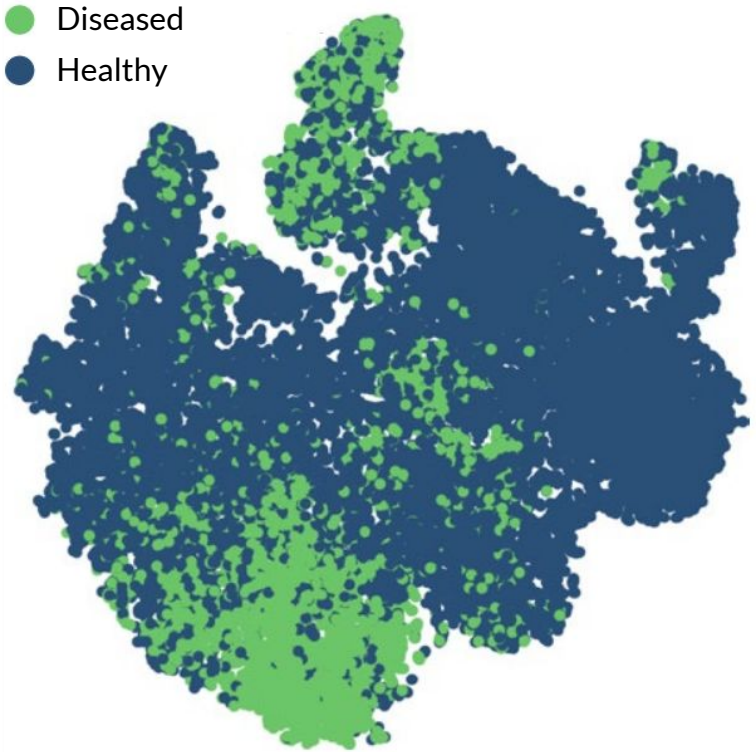
Reconstruction Loss

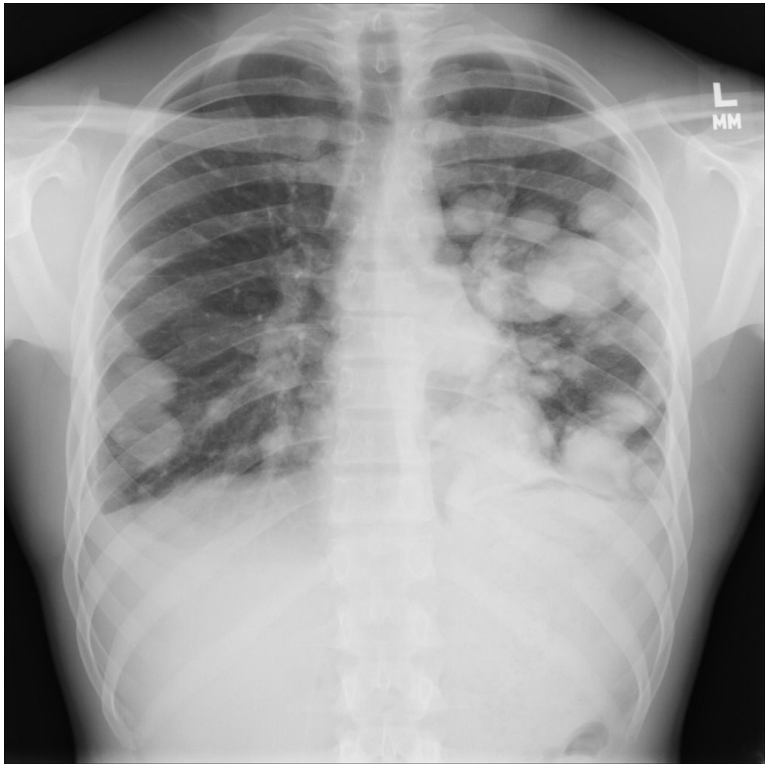
Encoder Matching

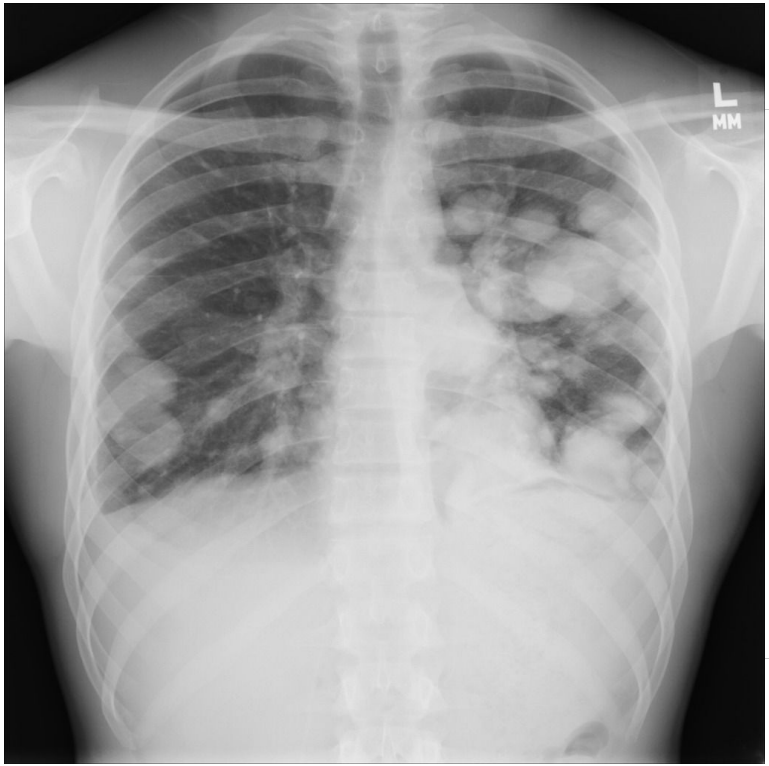
Discrimination



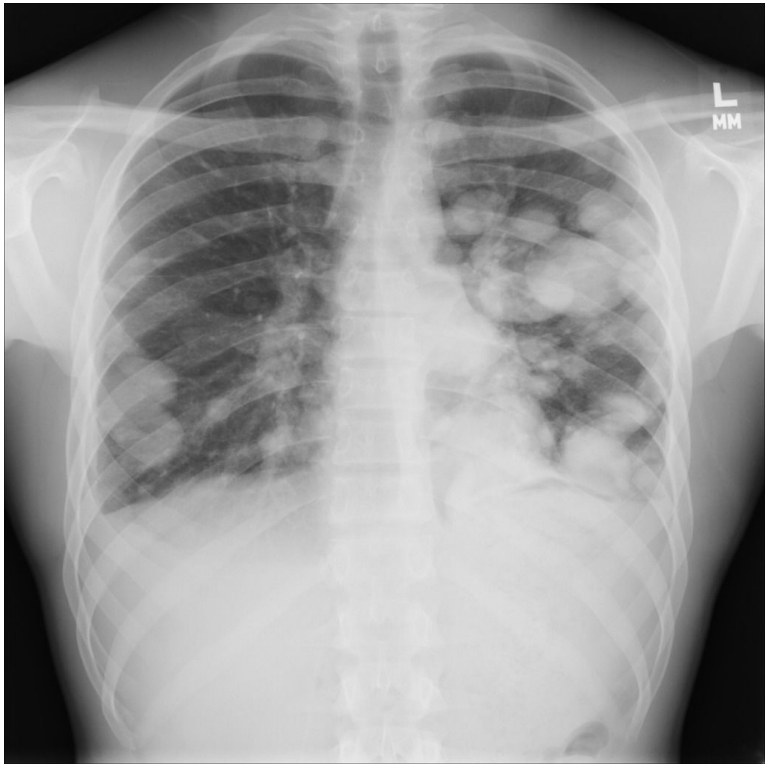



Diseased
Healthy

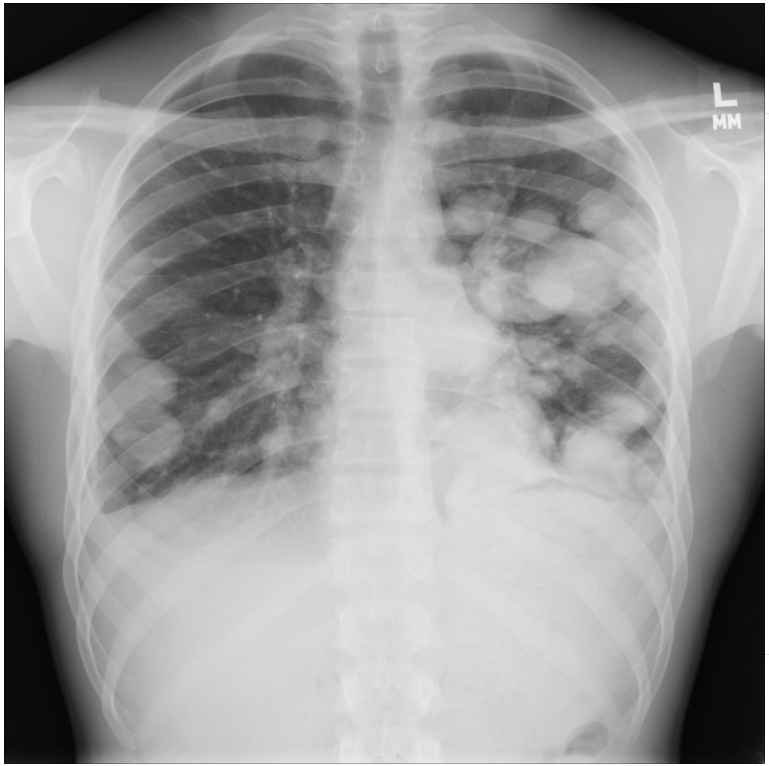

L
MM




```
def evaluate(image):
    reconstruction_loss = image_diff(image, reconstruct(image))
    embedding           = in_dist(healthy_embed, encode(image))
    discriminator       = netD(image)

    total_loss = reconstruction_loss + embedding + discriminator

    threshold = 0.30

    return total_loss > threshold
```

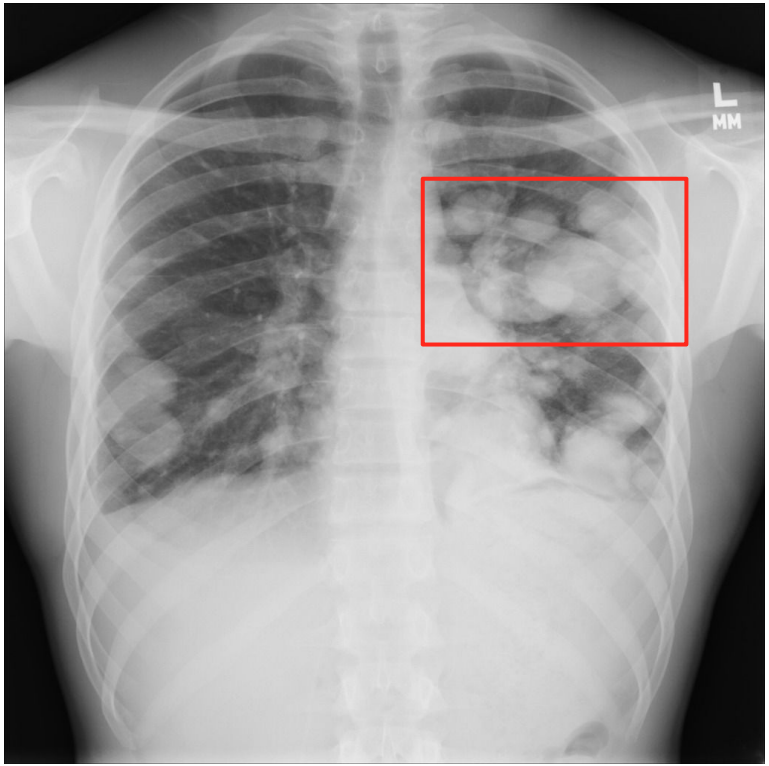


Anomaly



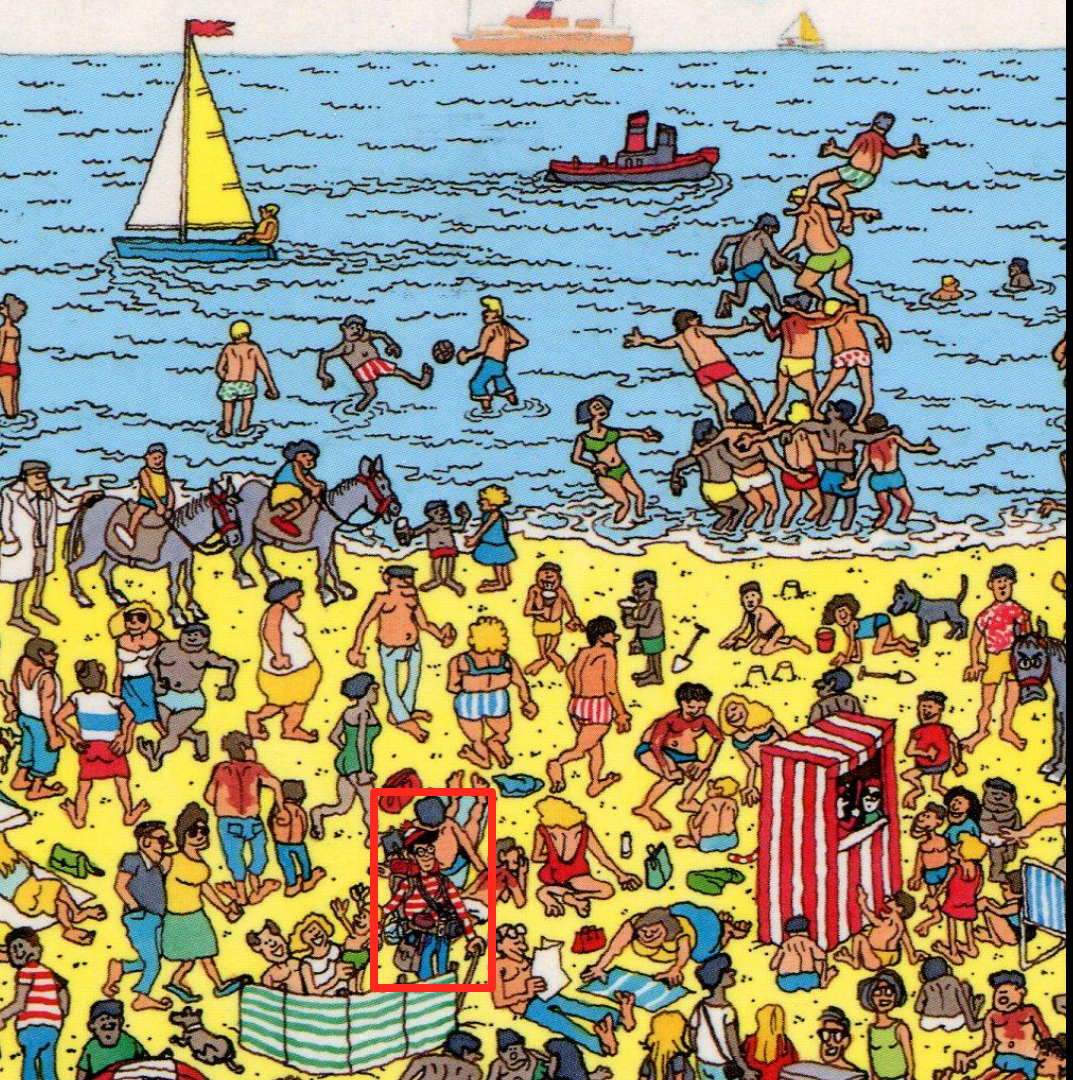
Anomaly

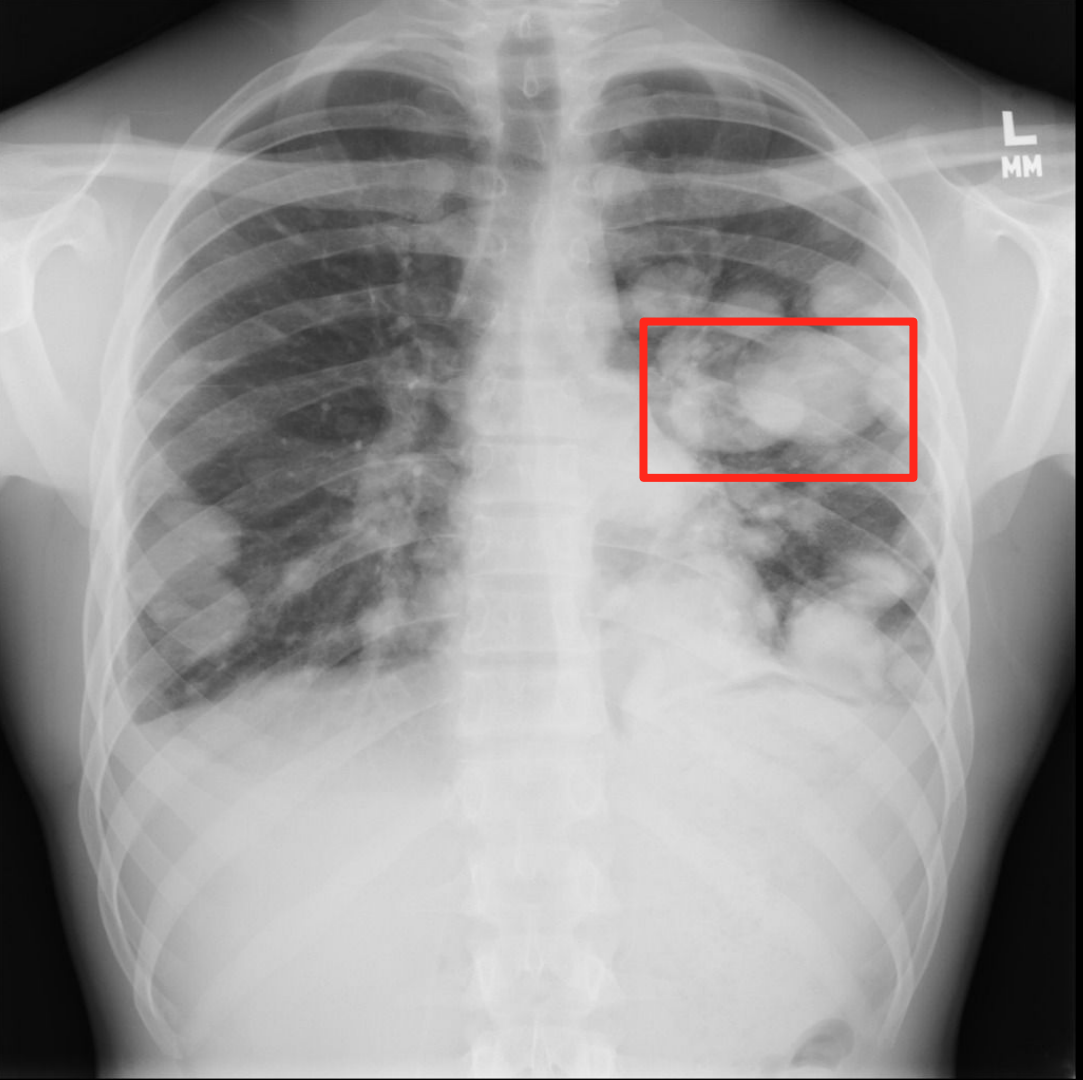
```
def localize_anomaly(image):
    reconstruction = reconstruct_image(image)

    image_similarity = similar(image, reconstruction)
    anomaly_location = local_min(image_similarity)

    return anomaly_location
```



Anomaly

ZILC




L
MM

liamhinzman.com

liamhinzman@gmail.com