In [1]:
```
import tensorflow as tf
import os
import numpy as np
from matplotlib import pyplot as plt
```

In [2]:
```
path=r"C:\Users\vanam\Downloads\archive (82)covid_train_data\Covid19-dataset\train"
```

In [3]:
```
print(os.path.exists(path))
```

```
True
```

In [4]:
```
train_data=tf.keras.preprocessing.image_dataset_from_directory(path,image_size=(128,128),batch_size=50)
```

```
Found 251 files belonging to 3 classes.
```

In [5]:
```
all_images=[]
all_classes=[]
```

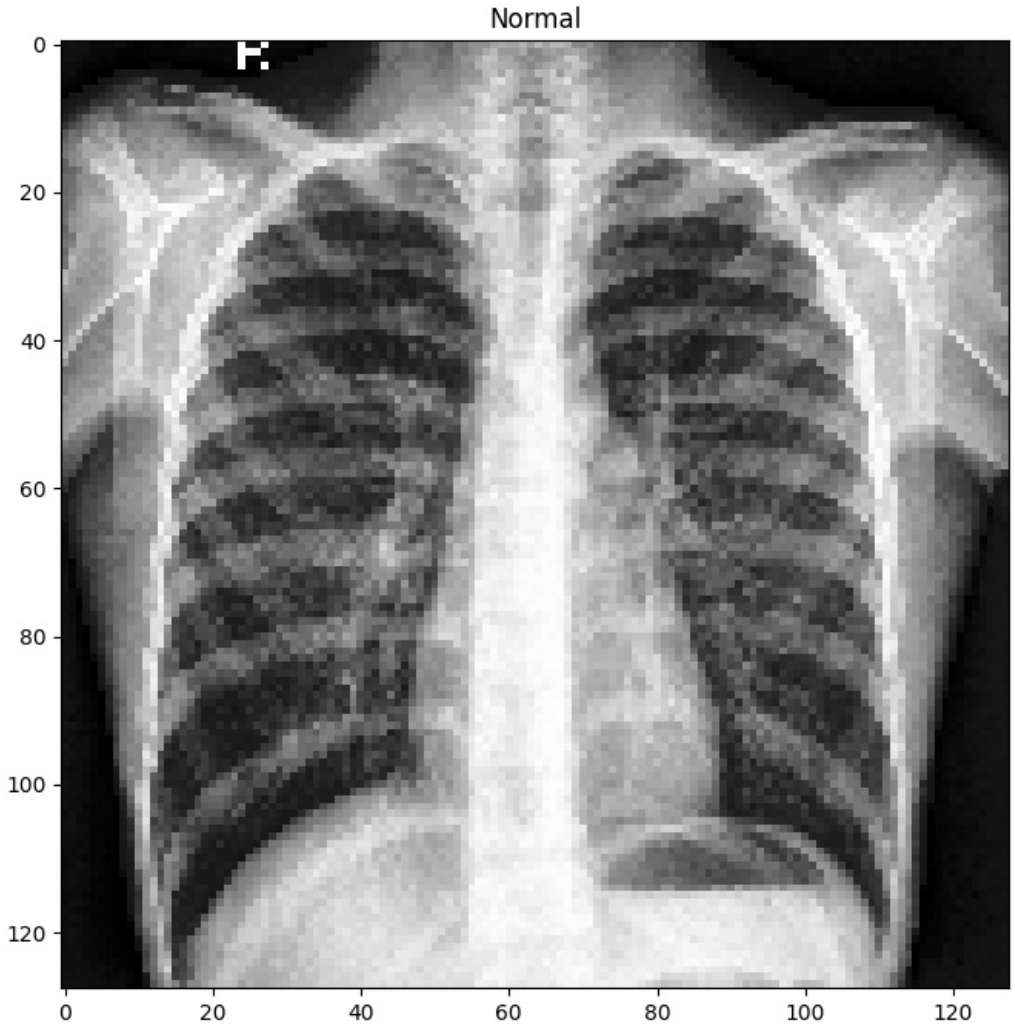
In [6]:
```
for image,label in train_data:
    all_images.append(image.numpy())
    all_classes.append(label)
```

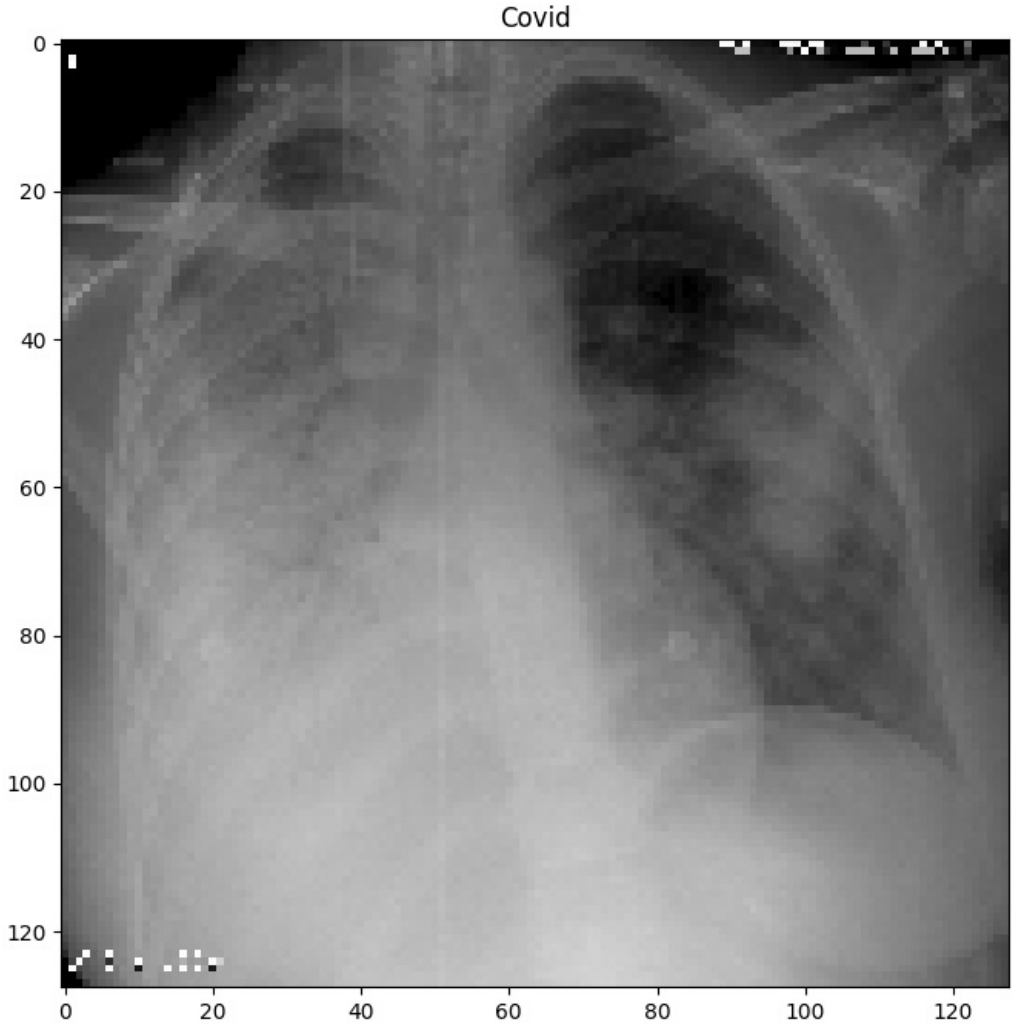
In [7]:
```
all_images=np.concatenate(all_images,axis=0)
all_classes=np.concatenate(all_classes,axis=0)
```

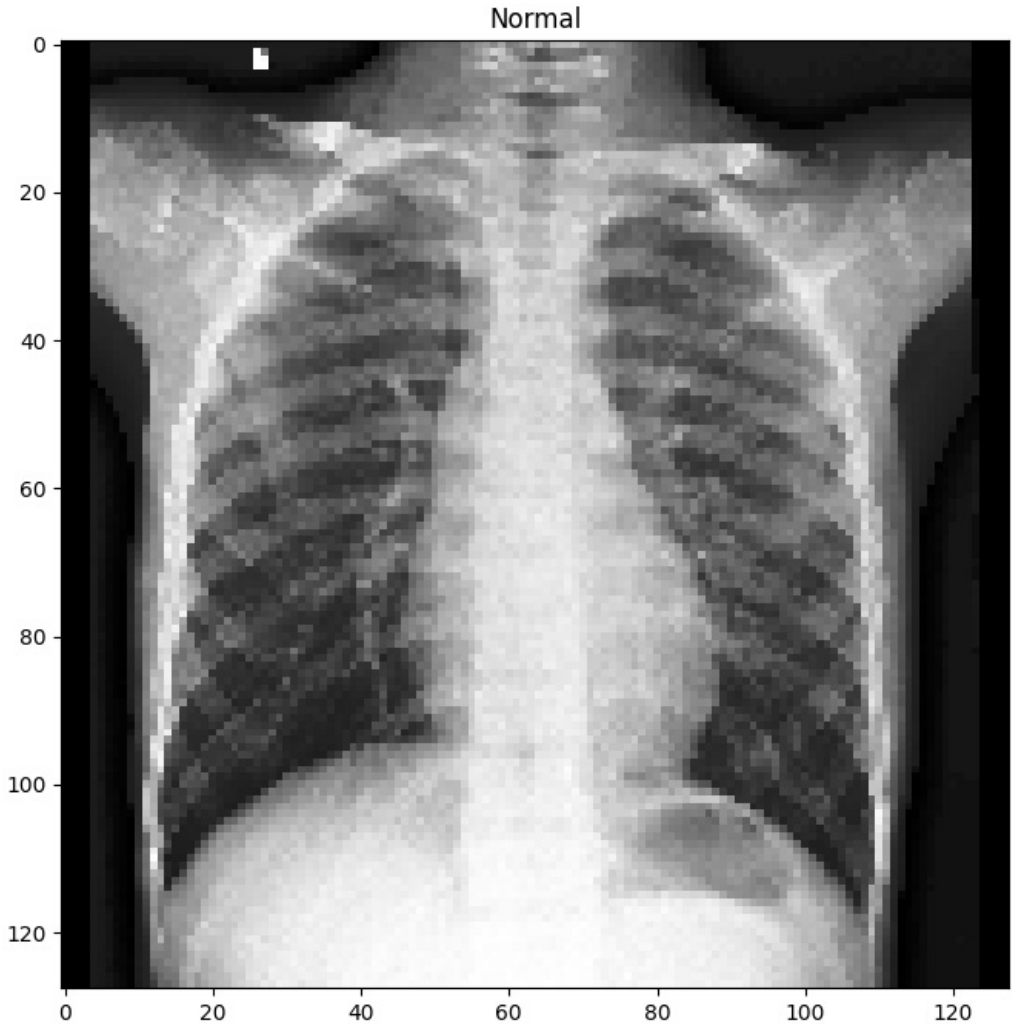
In [8]:
```
#Normalizing
all_images=all_images/255
```

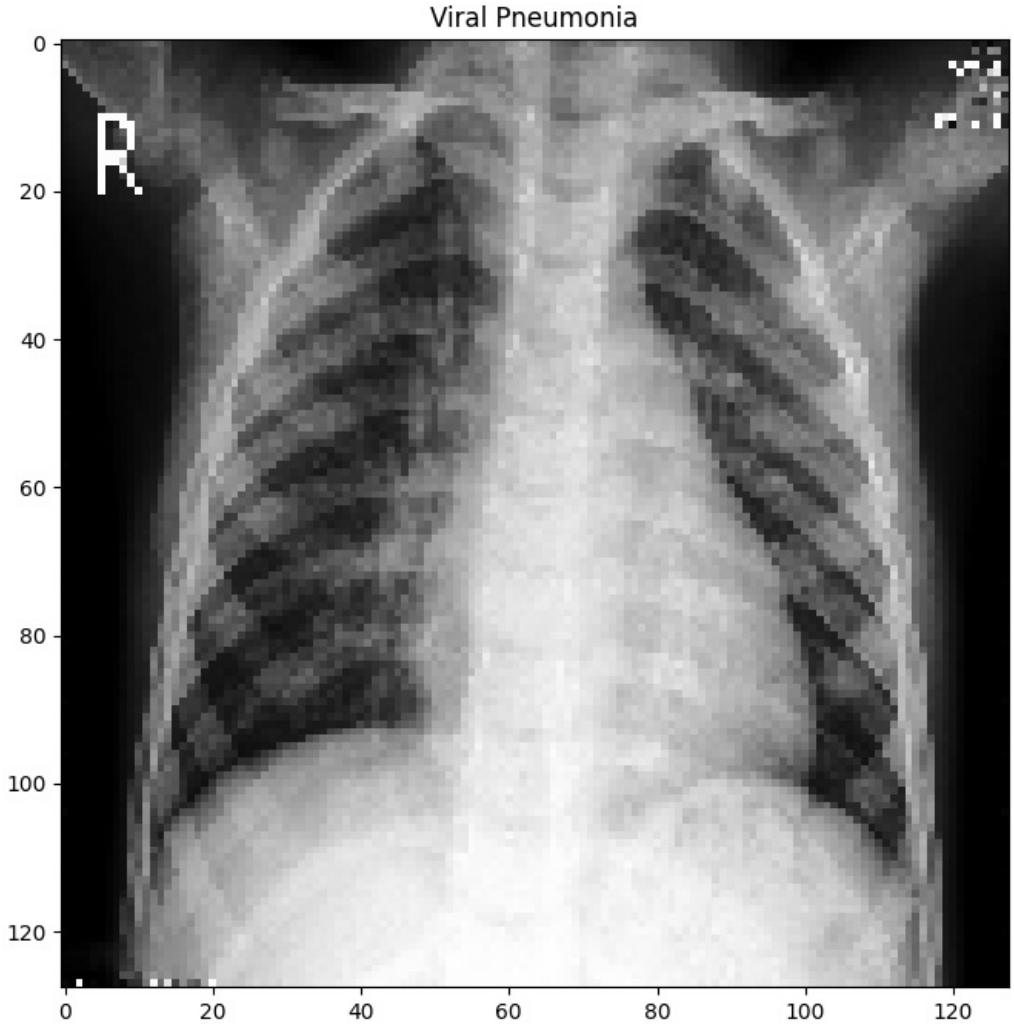
In [9]:
```
labeling={0:"Covid",1:"Normal",2:"Viral Pneumonia"}
```

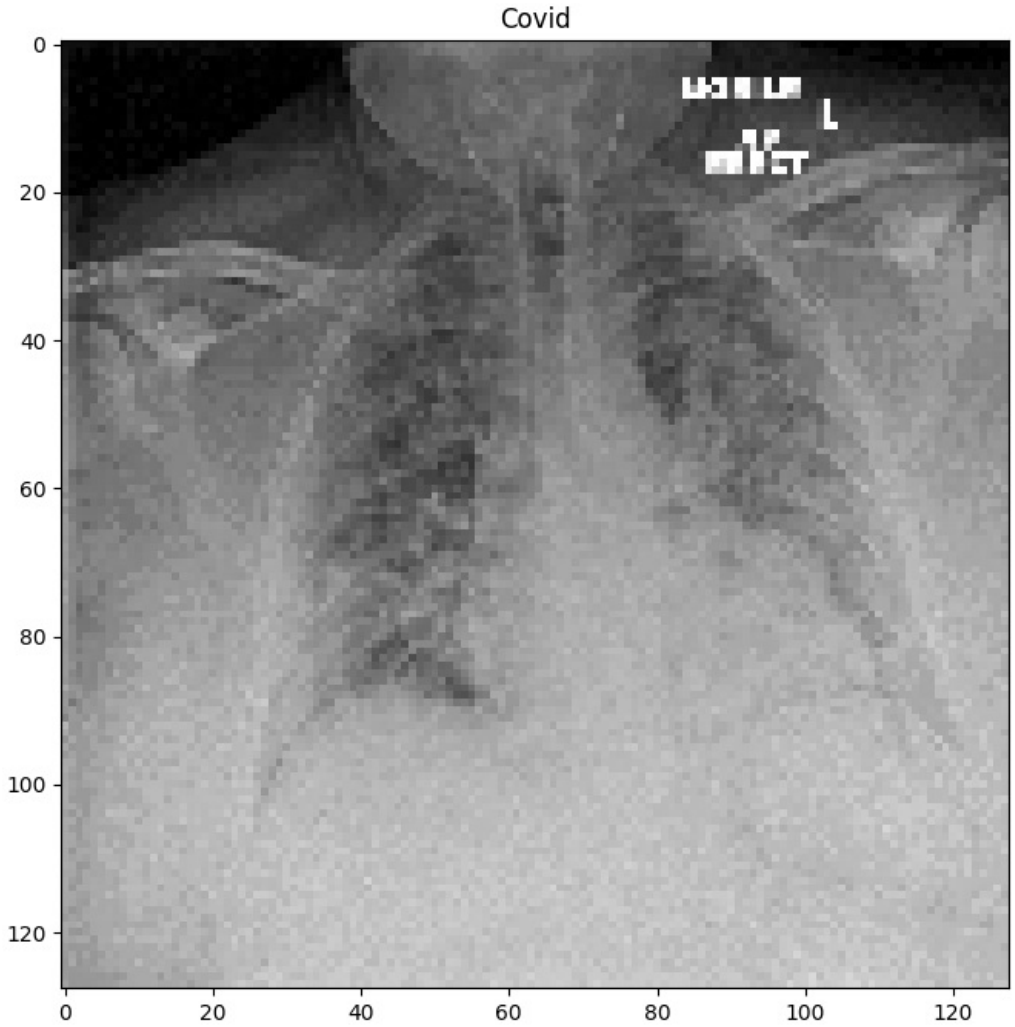
In [10]:
```
#visualizing first 30 images
for i in range(30):
    plt.figure(figsize=(8,8))
    plt.title(labeling[all_classes[i]])
    plt.imshow(all_images[i])
    plt.show()
```

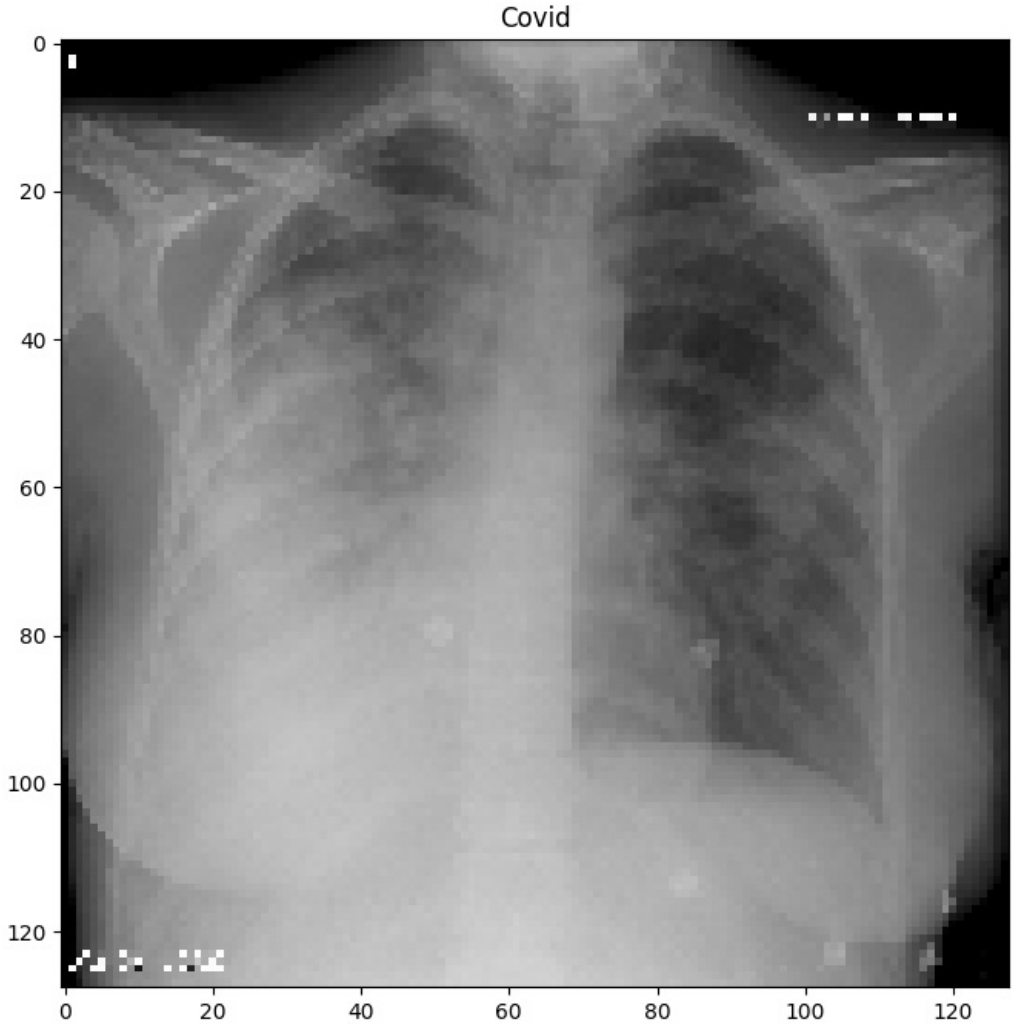


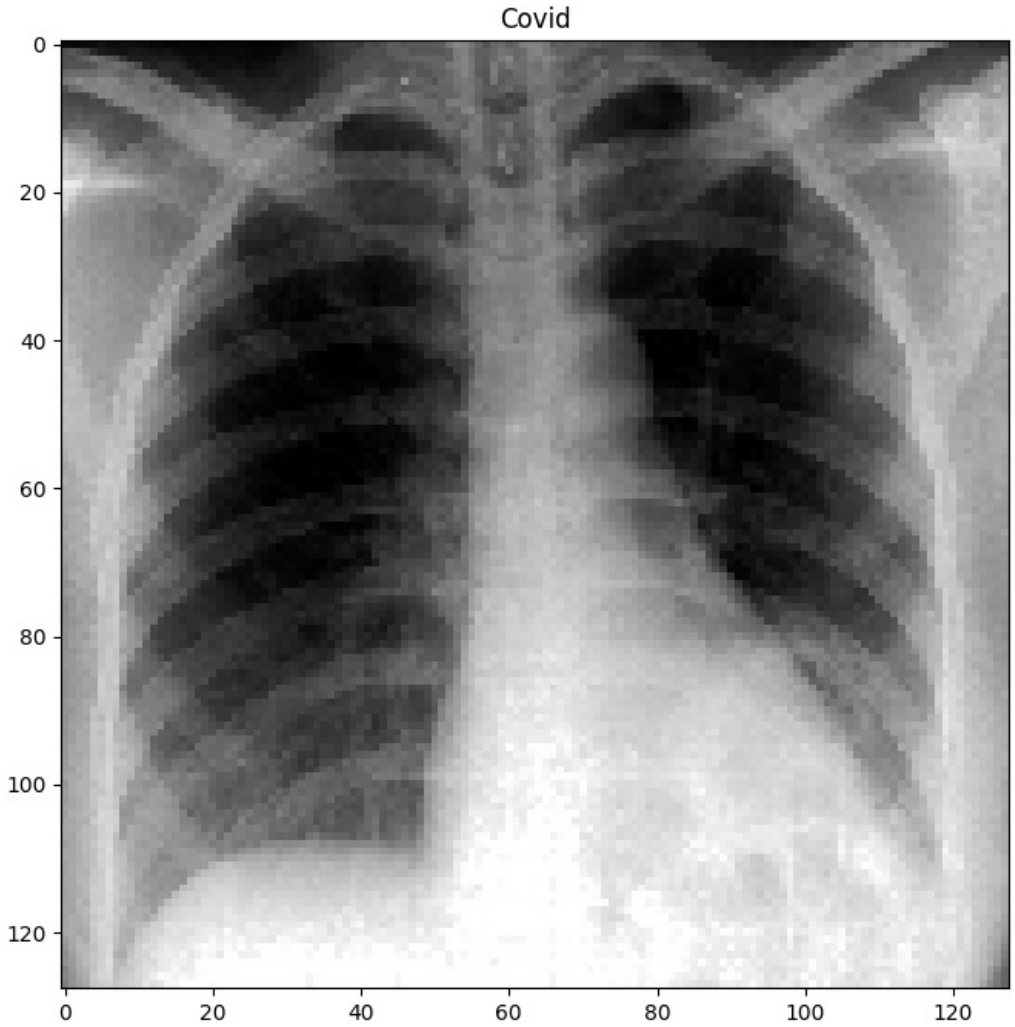

Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

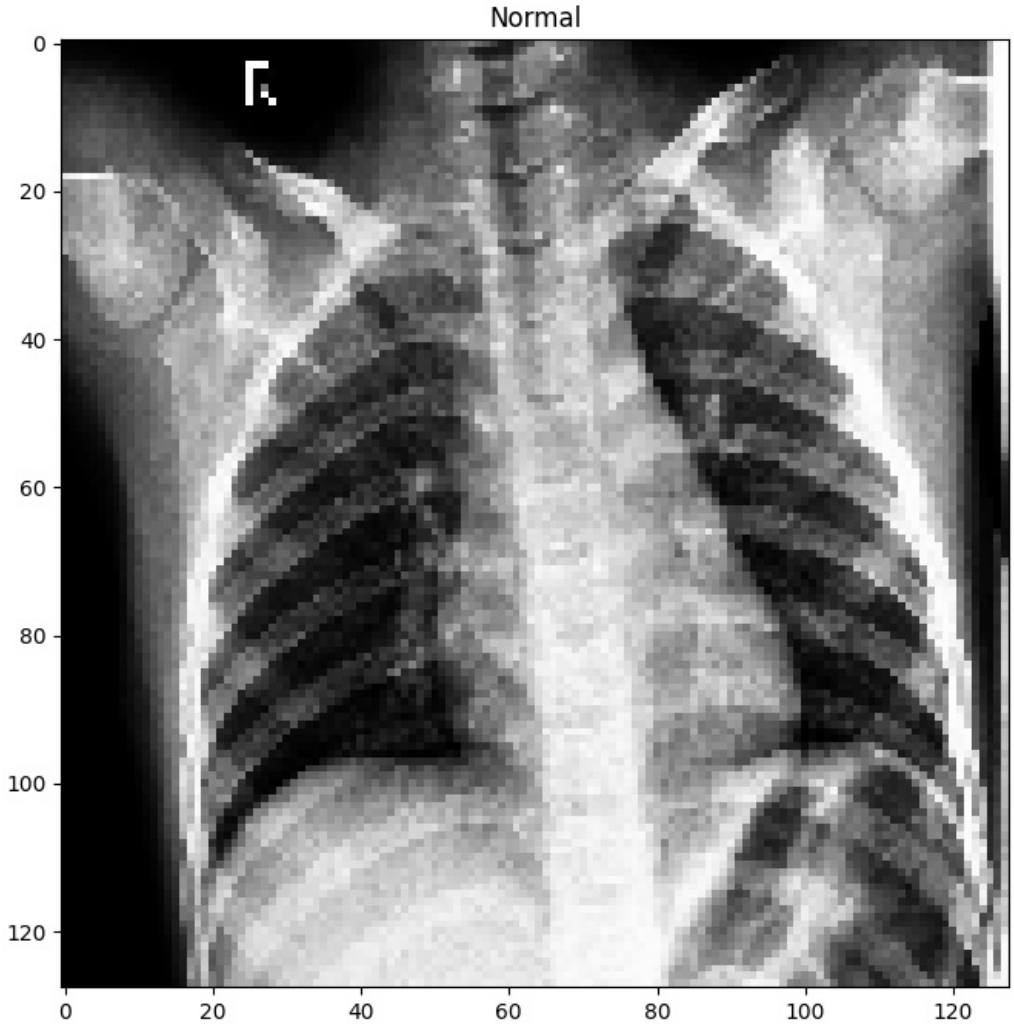

Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

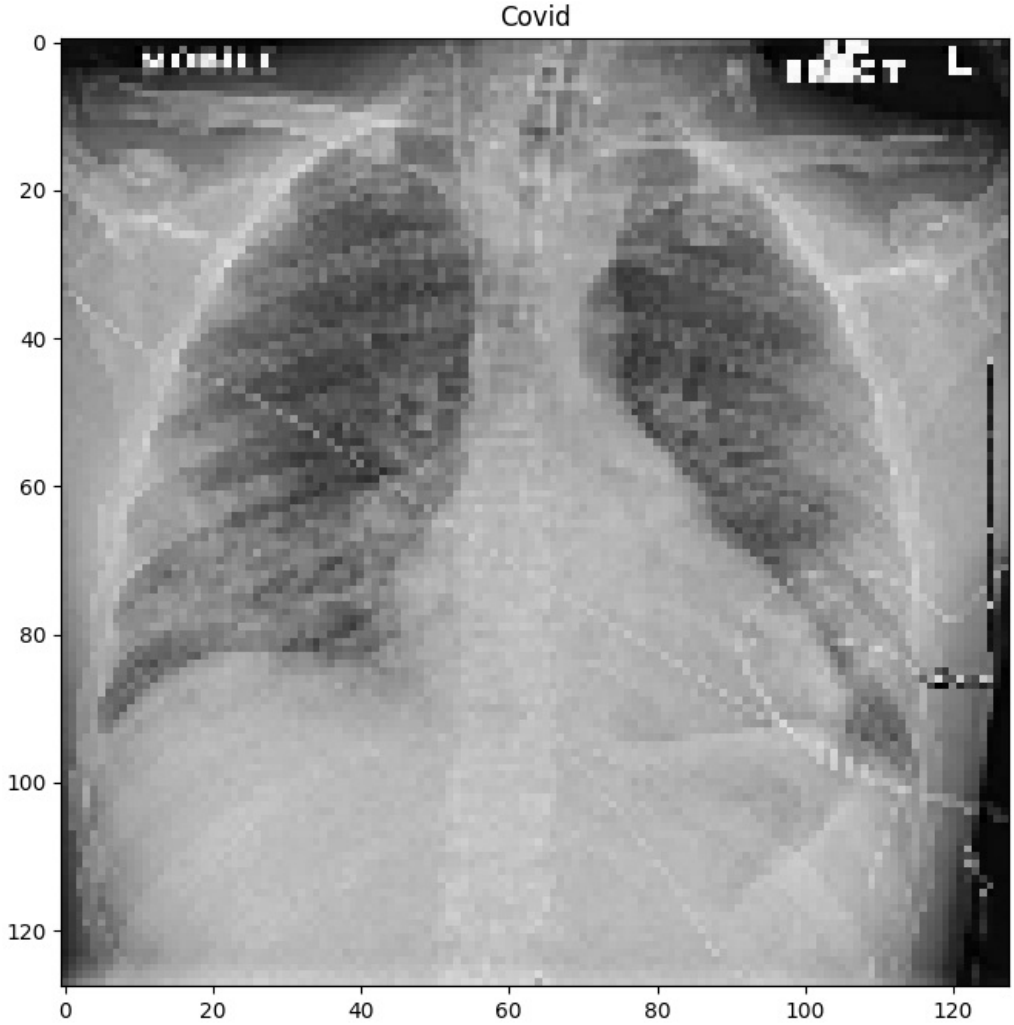

Viral Pneumonia
R
0
20
40
60
80
100
120
0
20
40
60
80
100
120

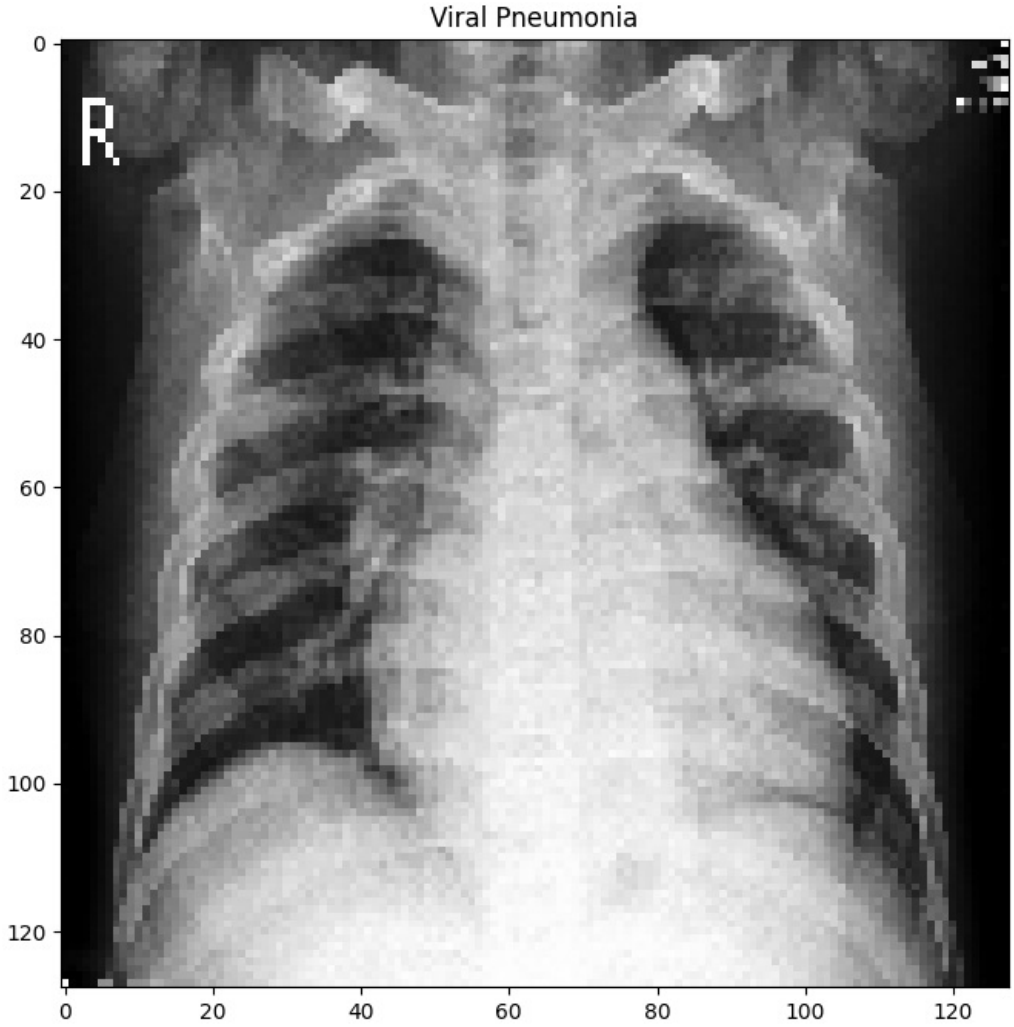

Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

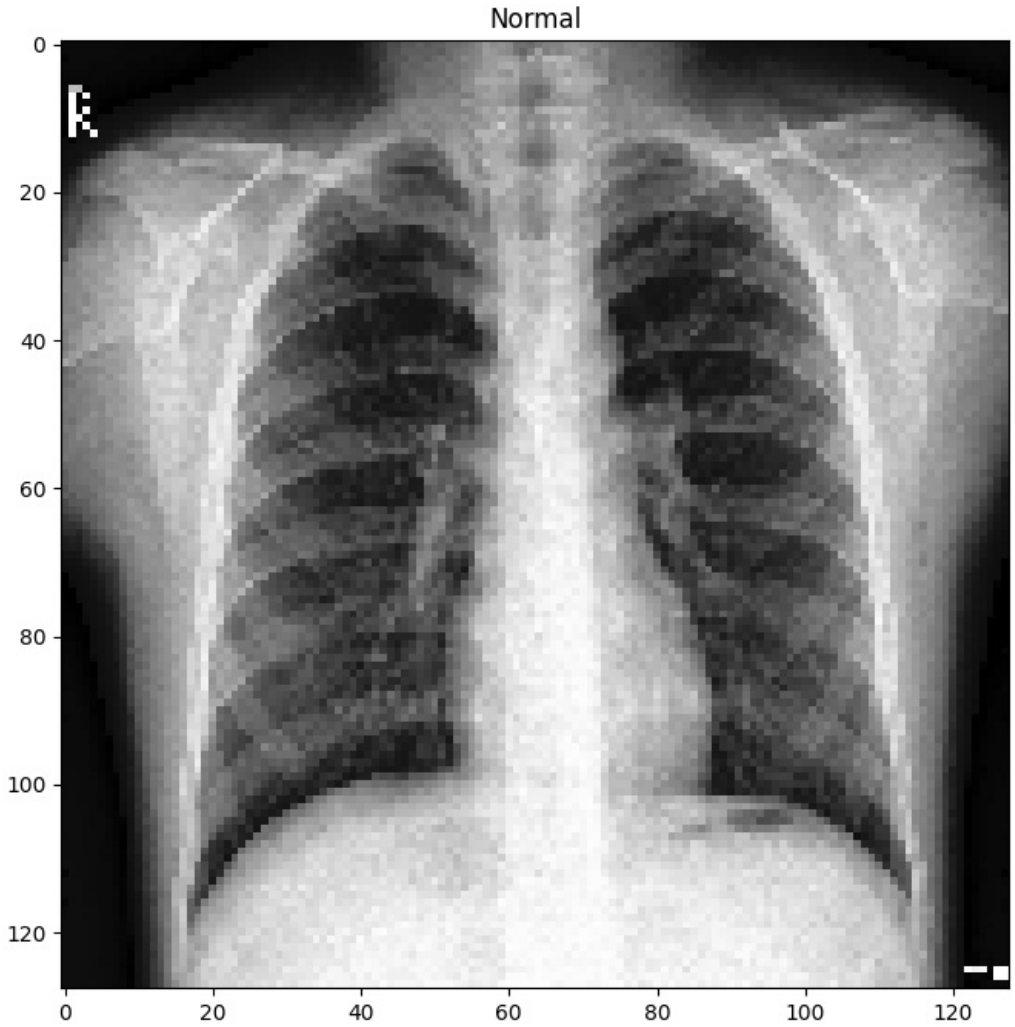

Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

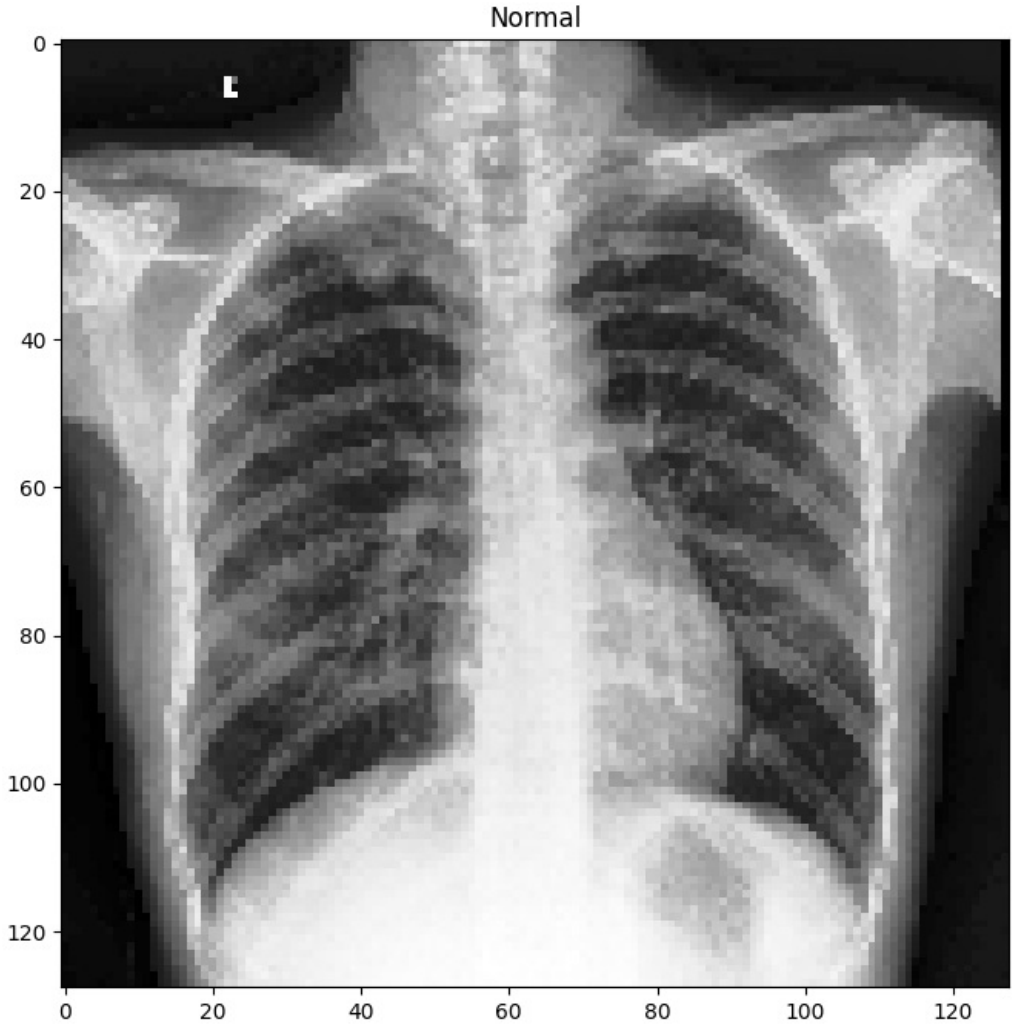

Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

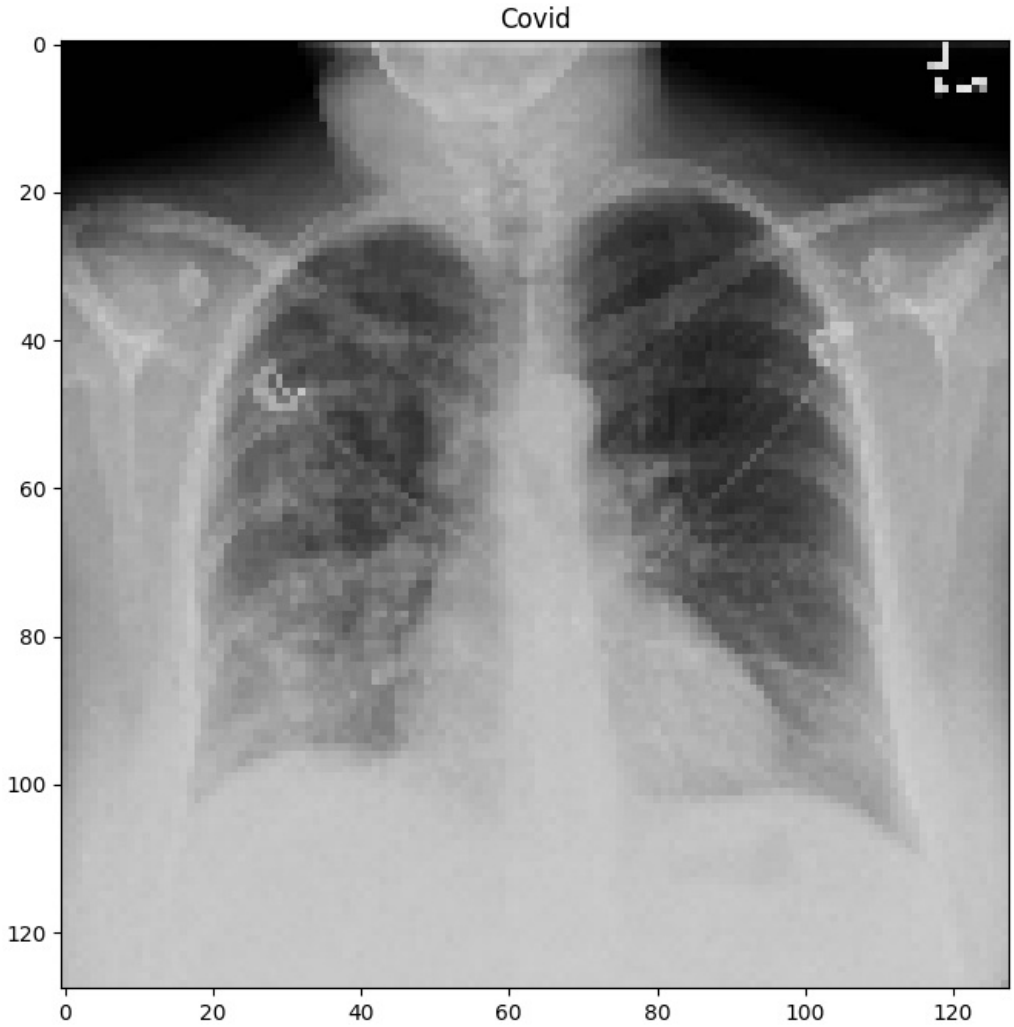

Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

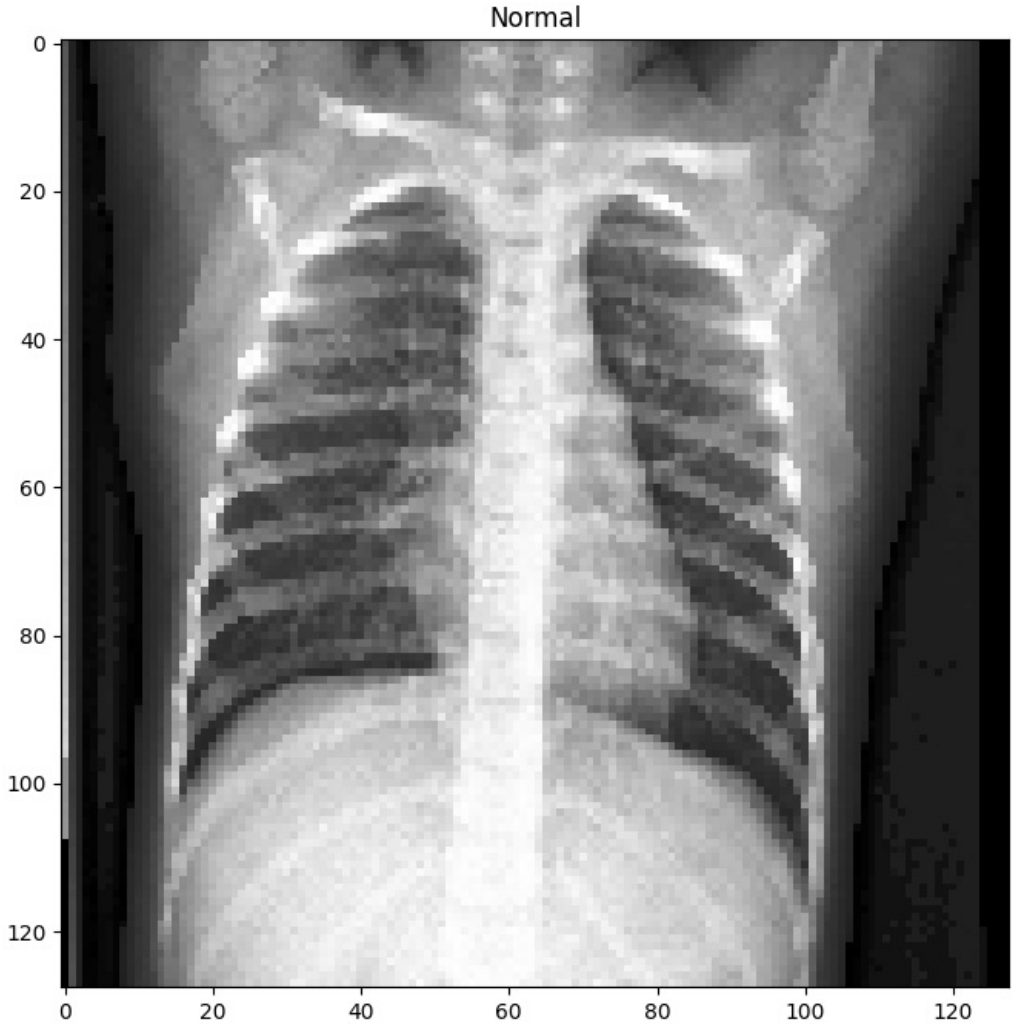

Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

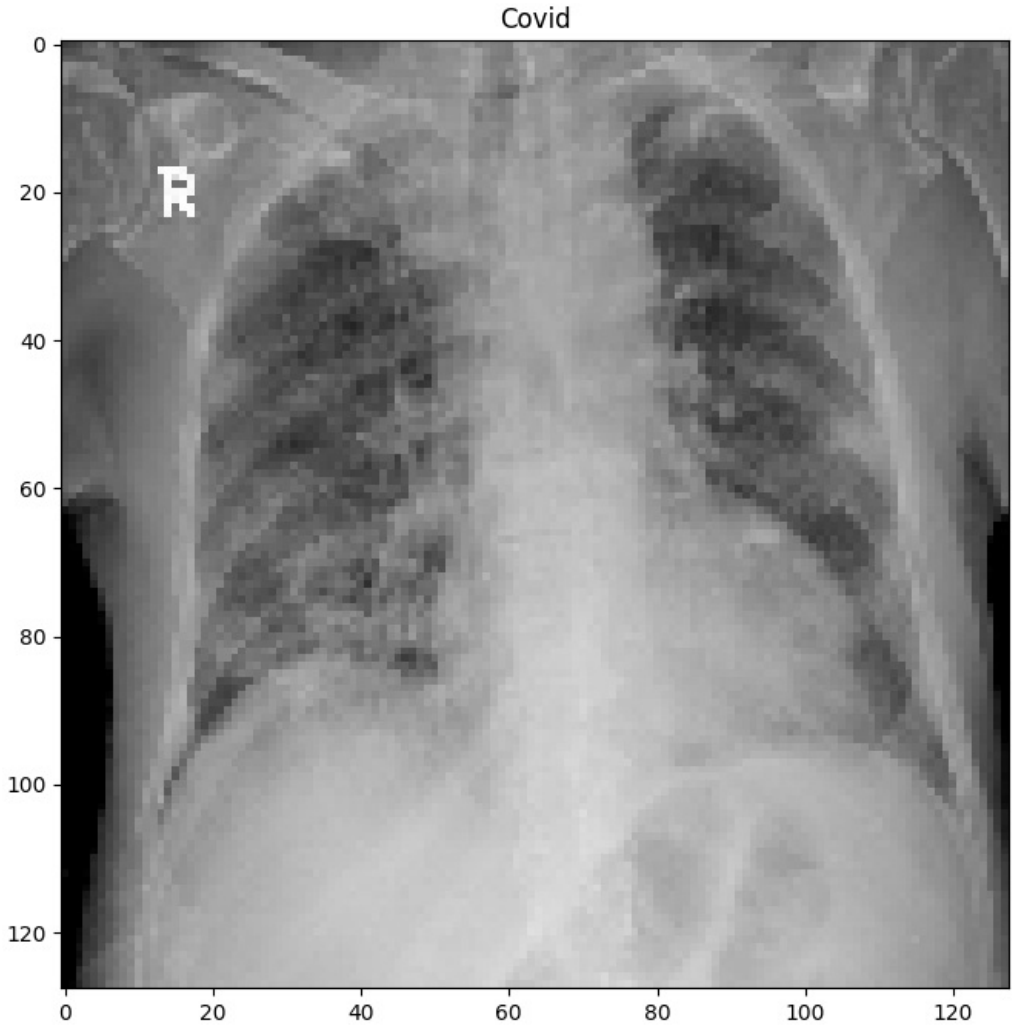

Viral Pneumonia
R
0
20
40
60
80
100
120
0
20
40
60
80
100
120

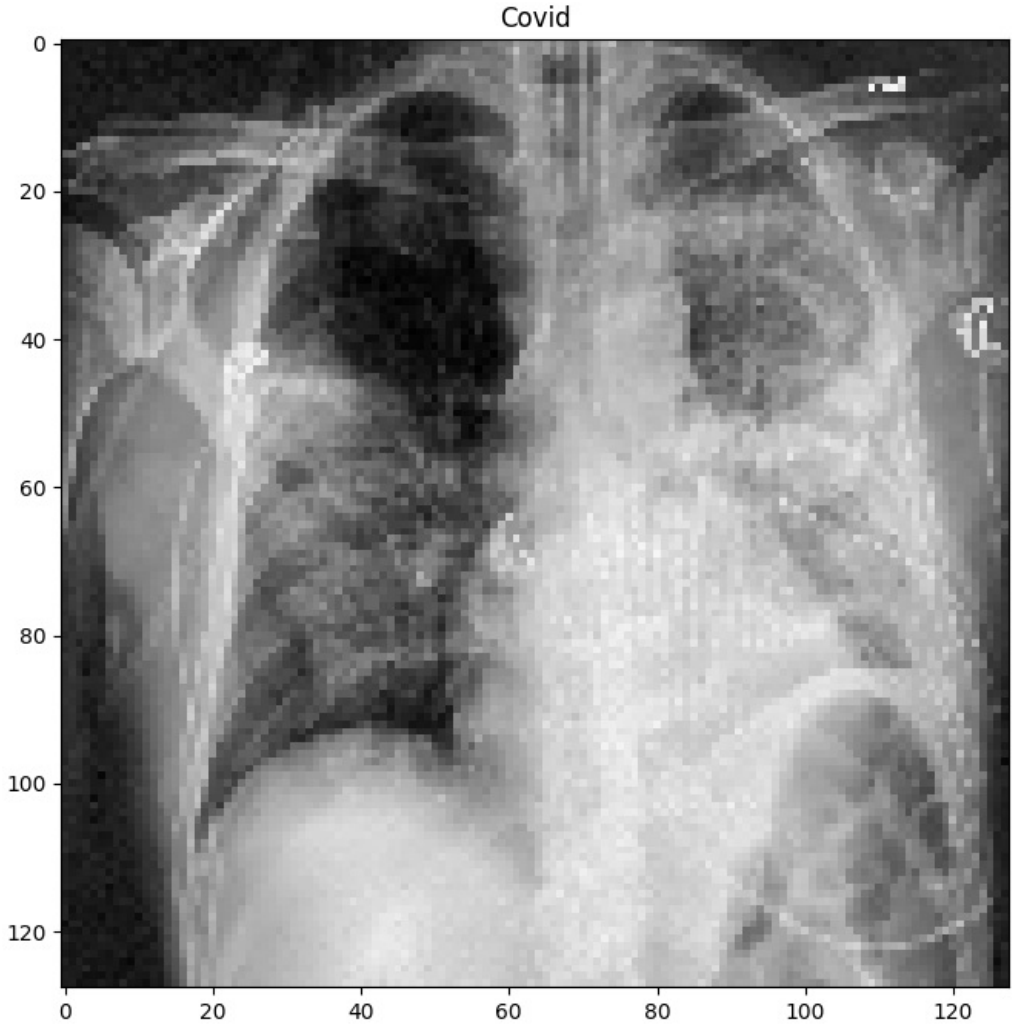

Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

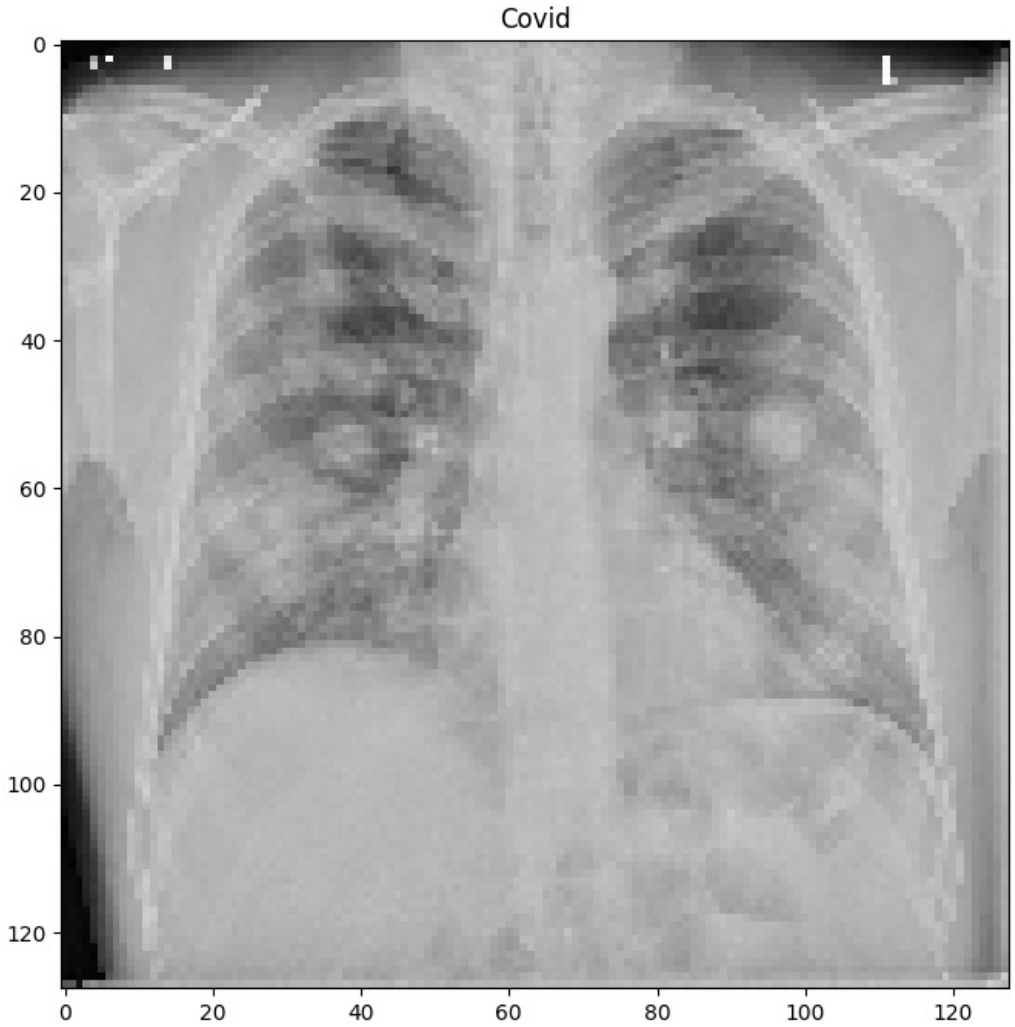

Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

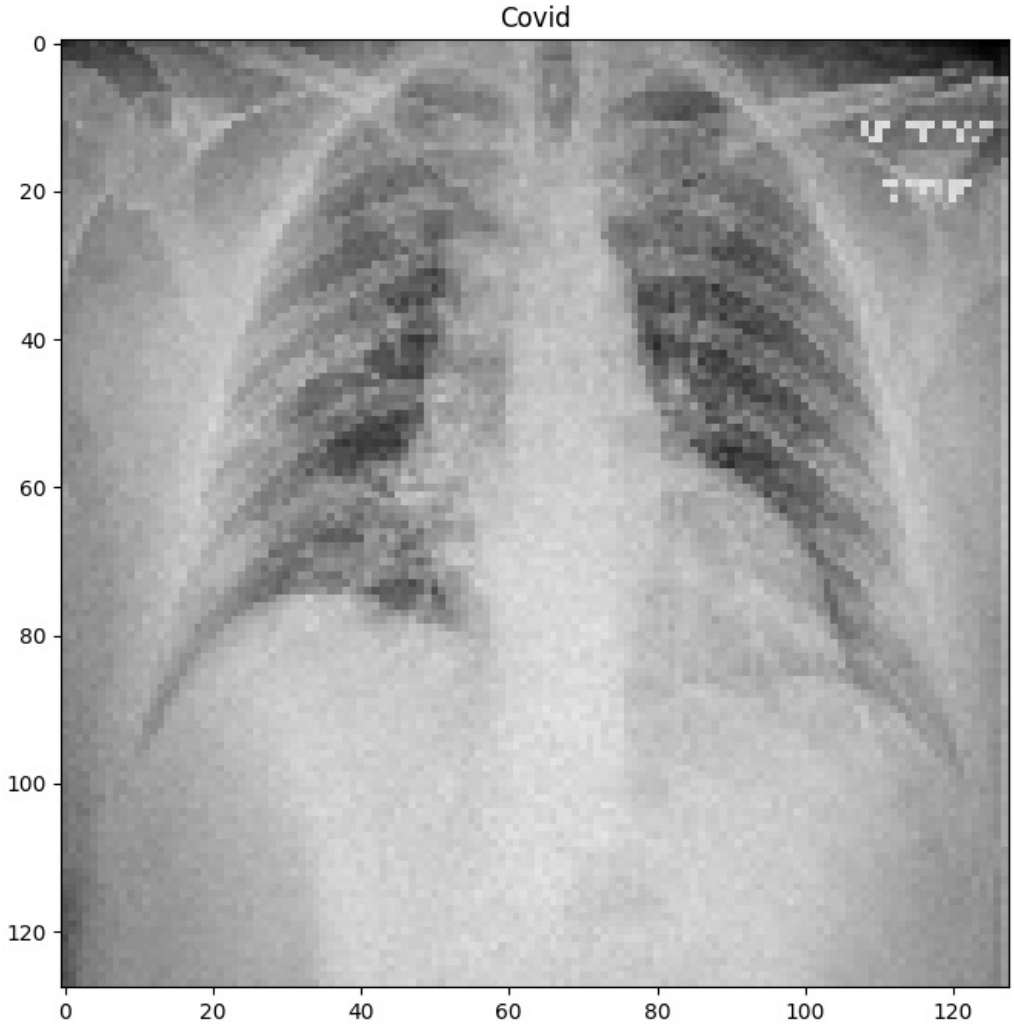

Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

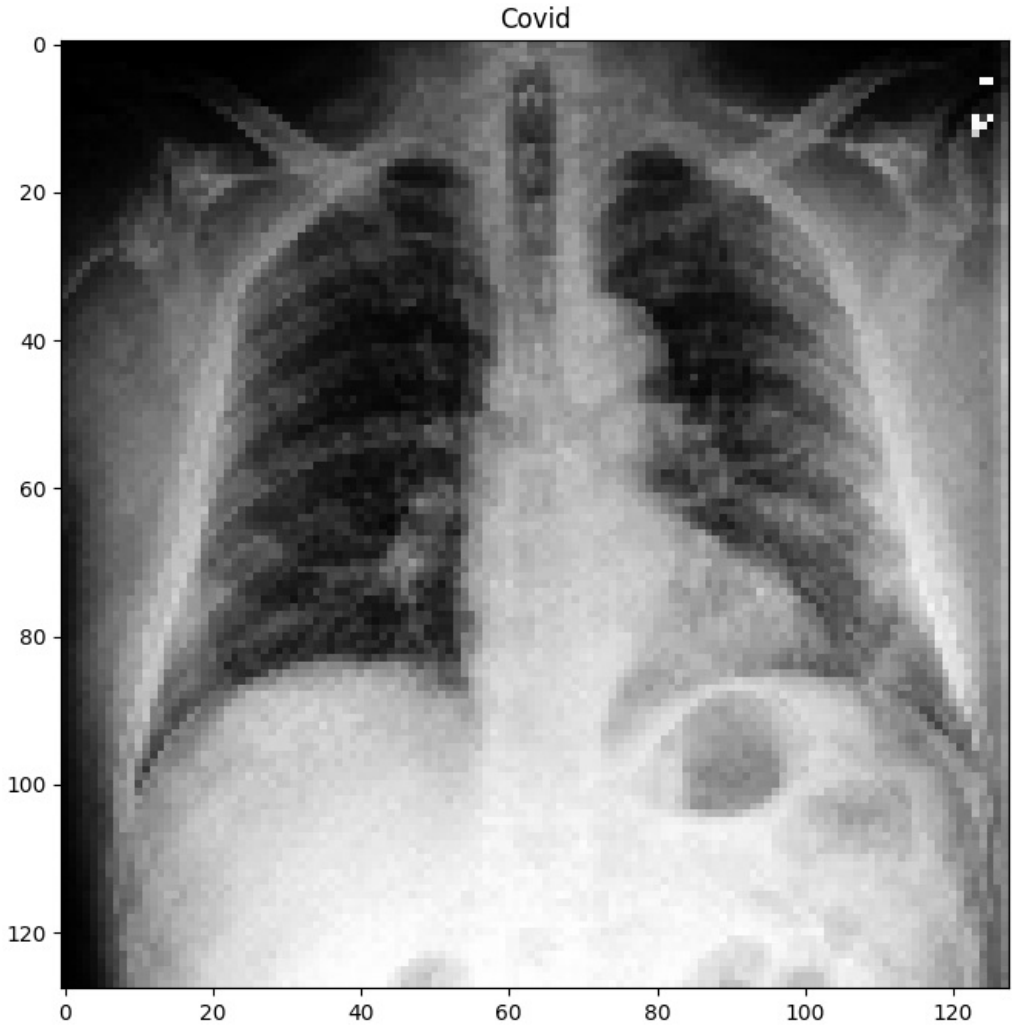

Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

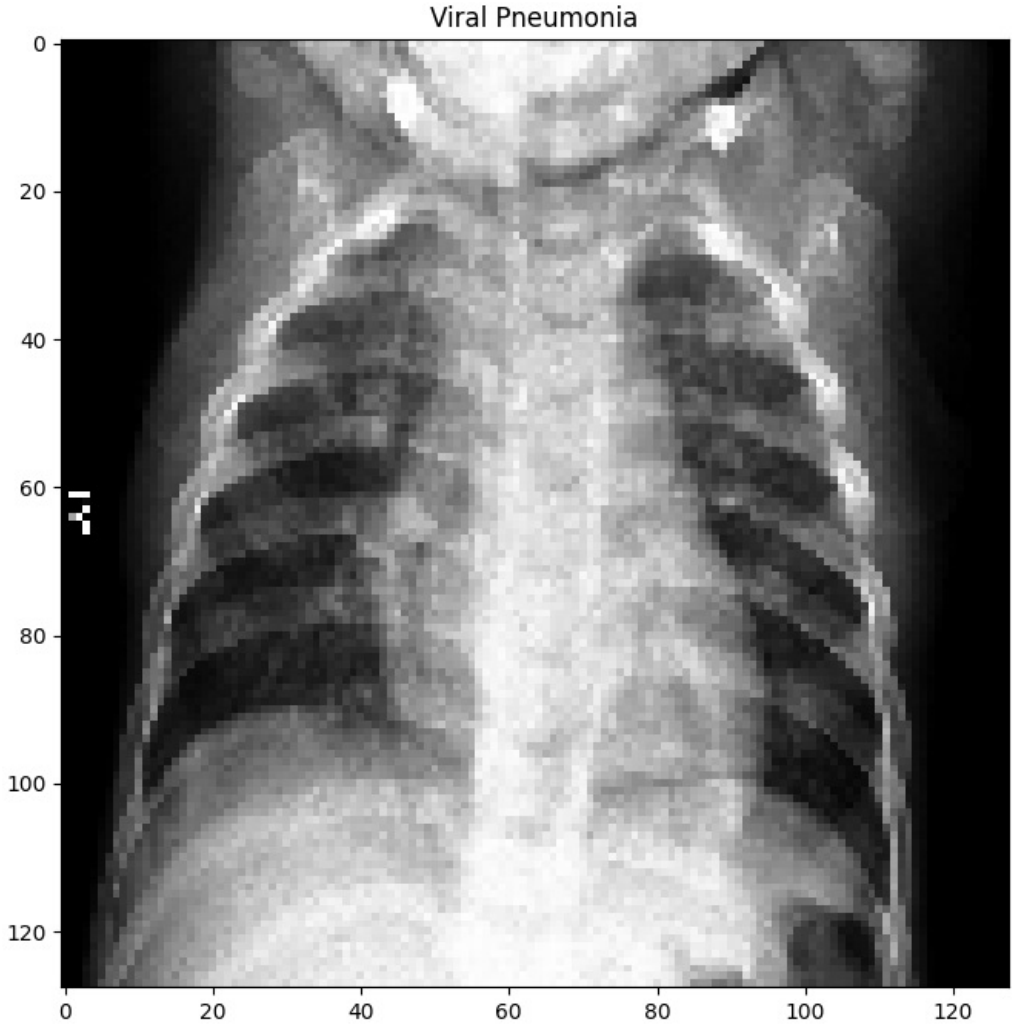

Covid
R
0
20
40
60
80
100
120
0
20
40
60
80
100
120

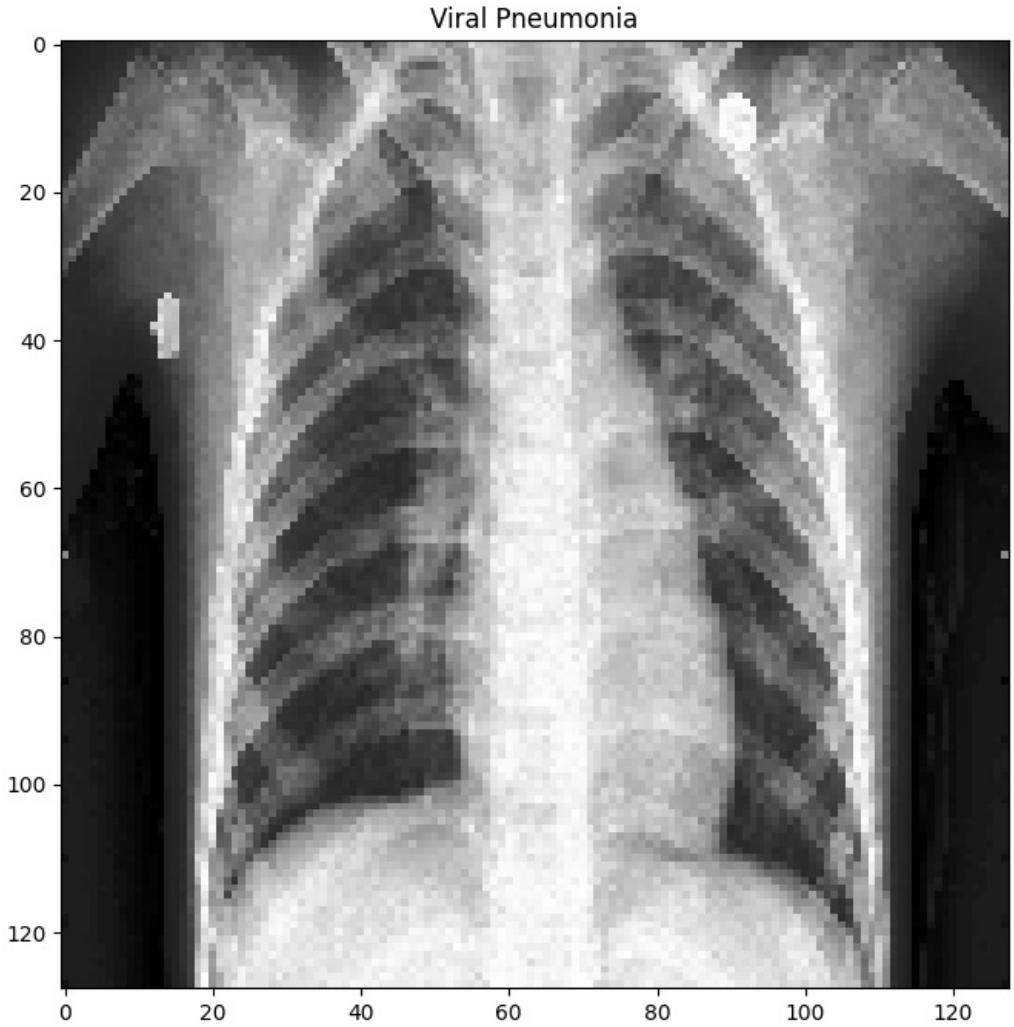

Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

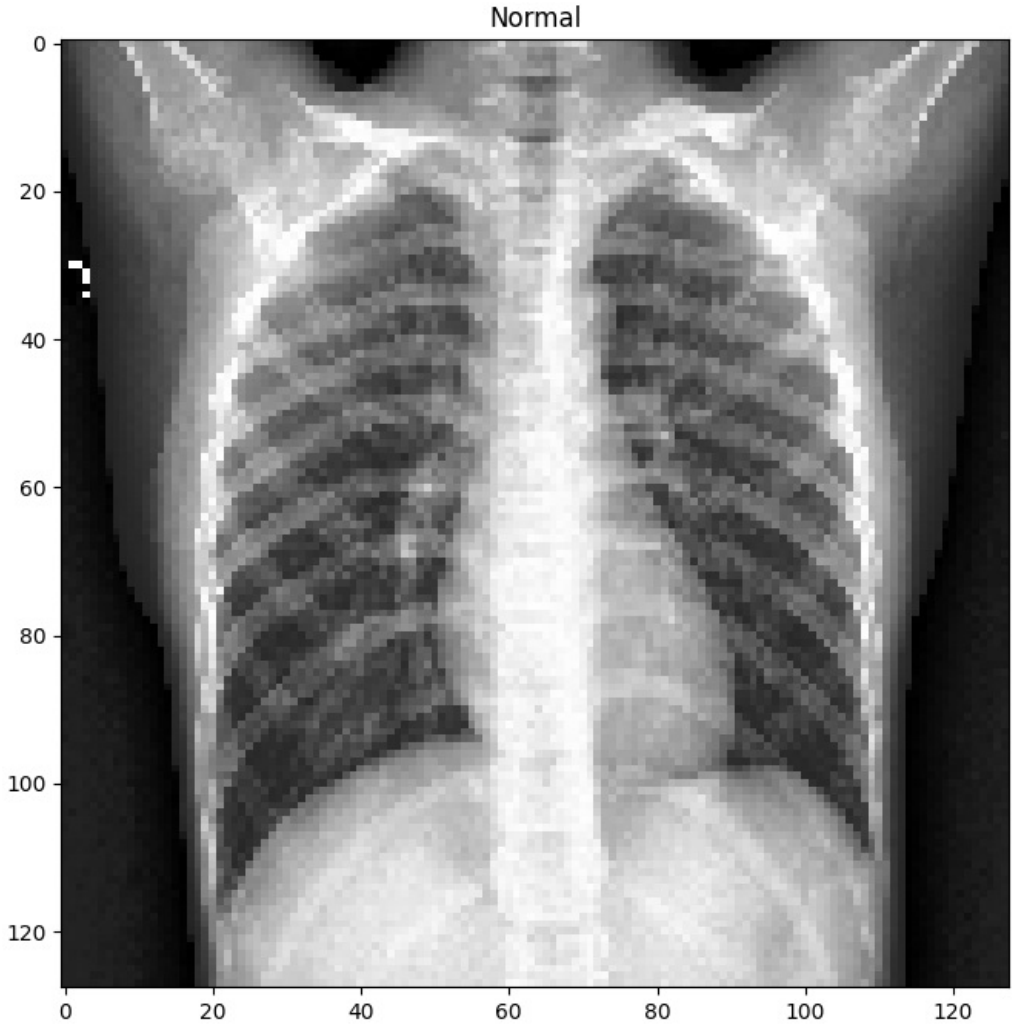

Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

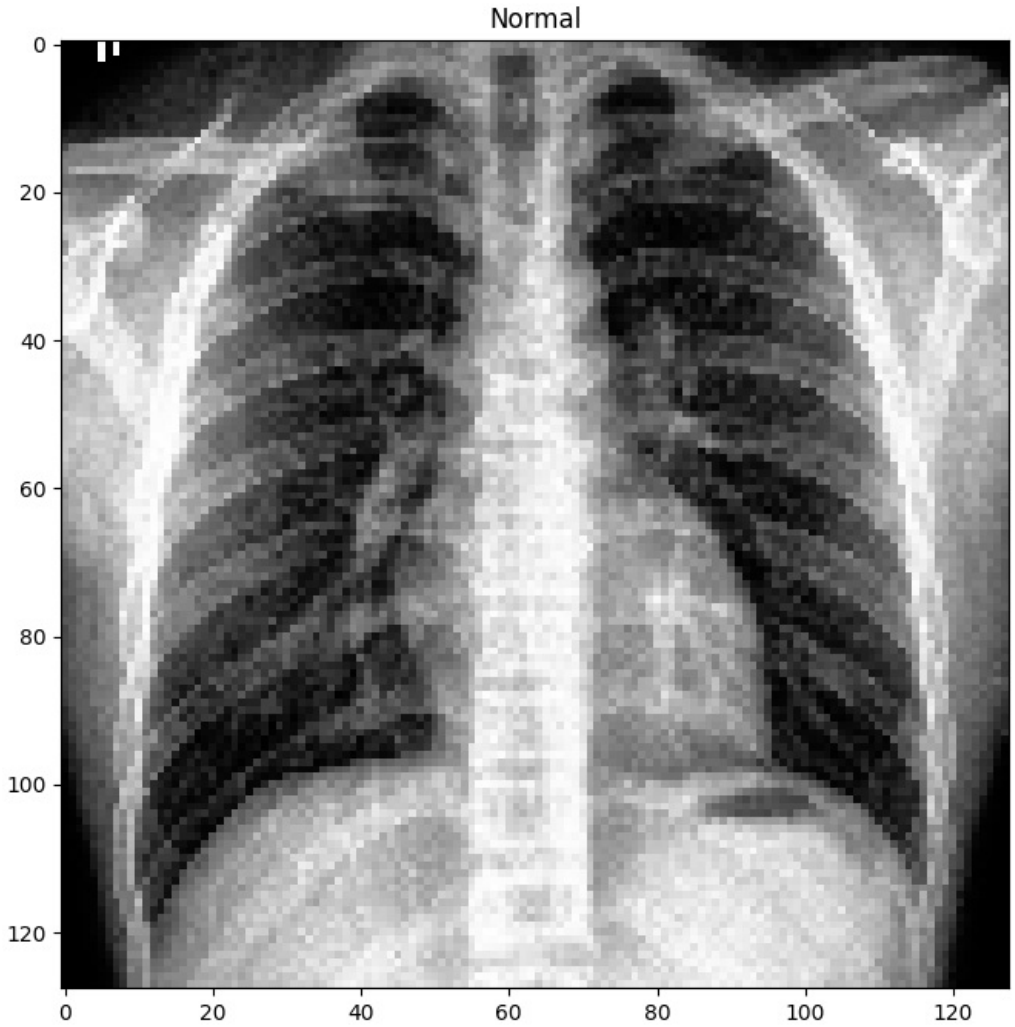

Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

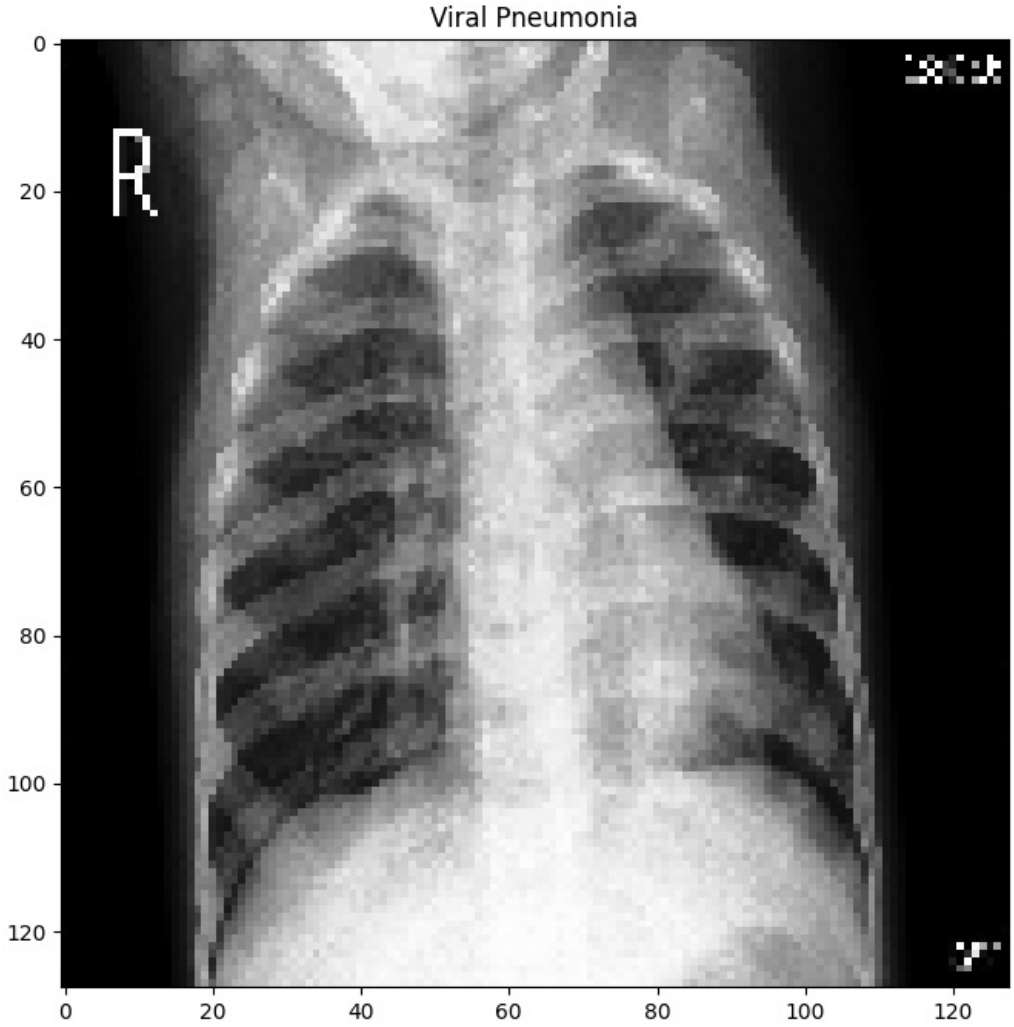

Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

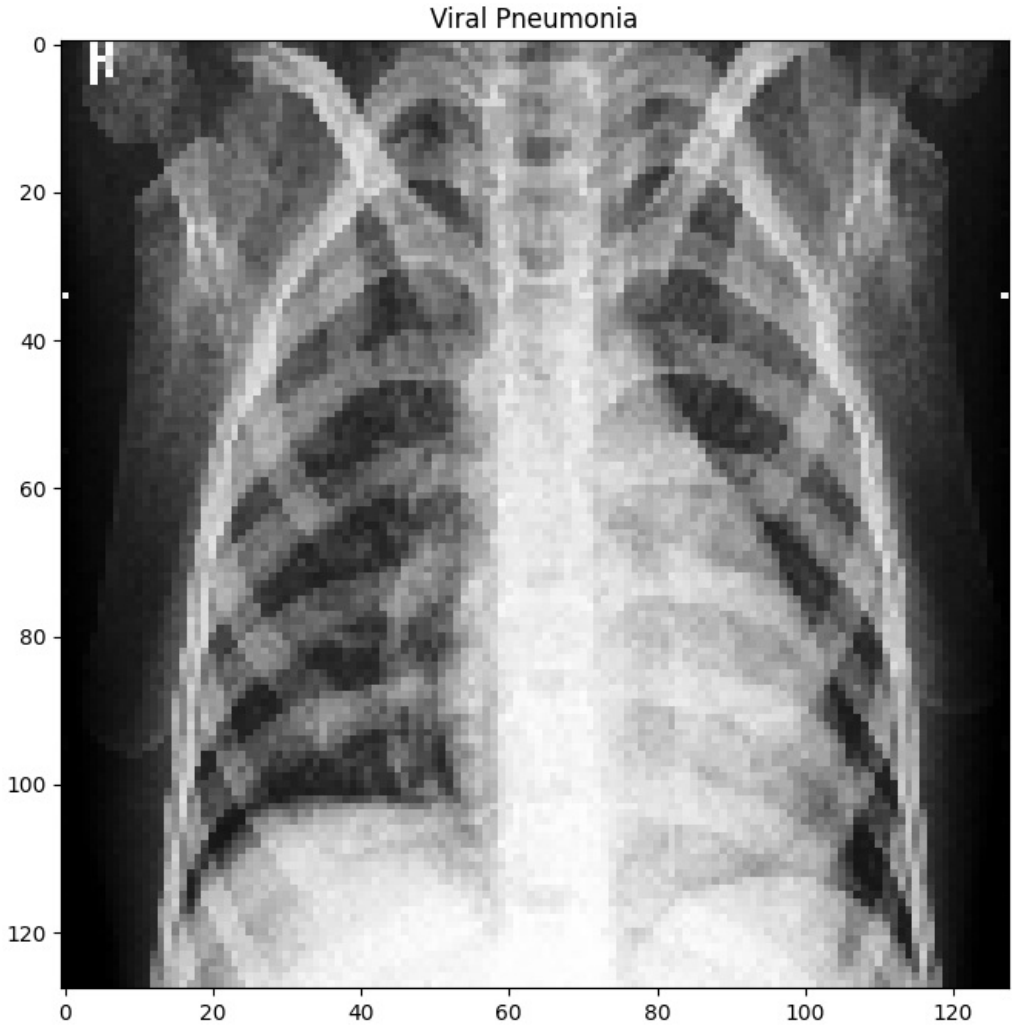

Viral Pneumonia
0
20
40
60
80
100
120
0
20
40
60
80
100
120

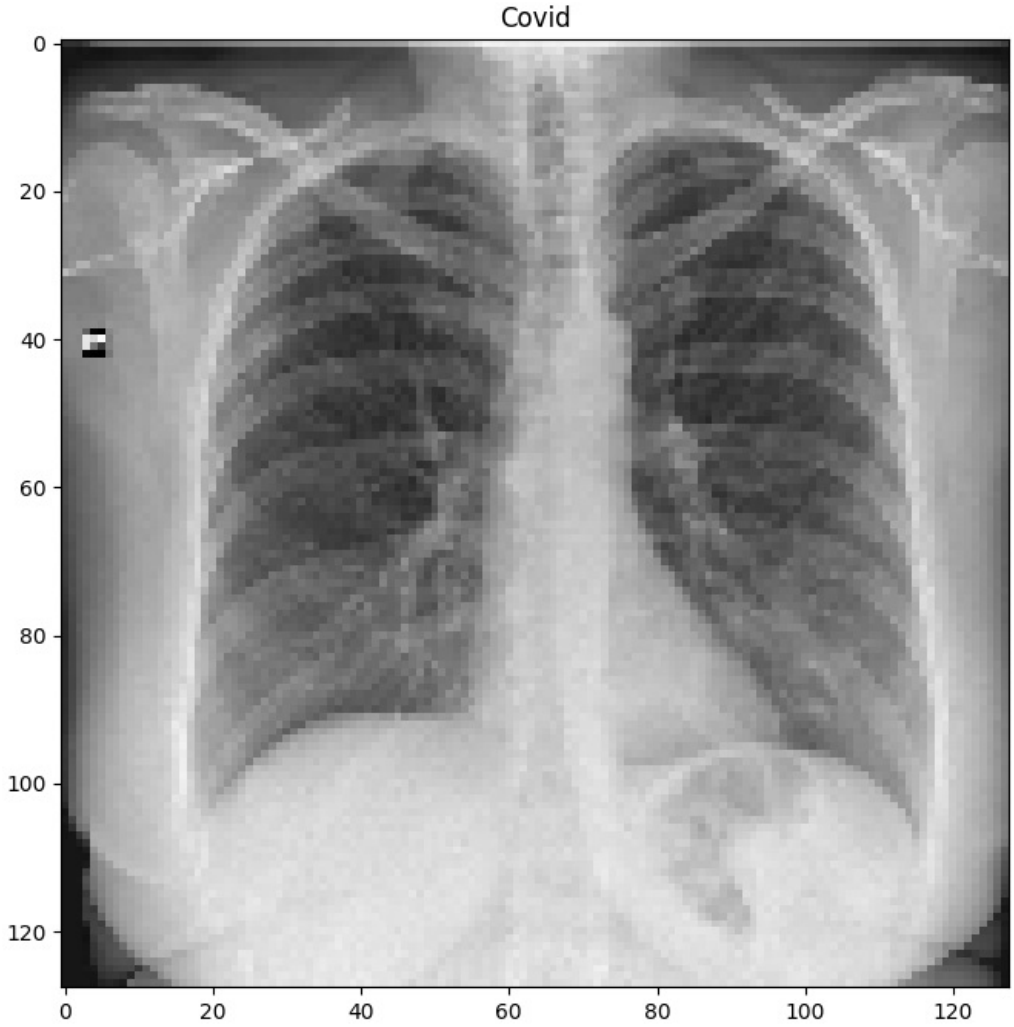

Viral Pneumonia
0
20
40
60
80
100
120
0
20
40
60
80
100
120

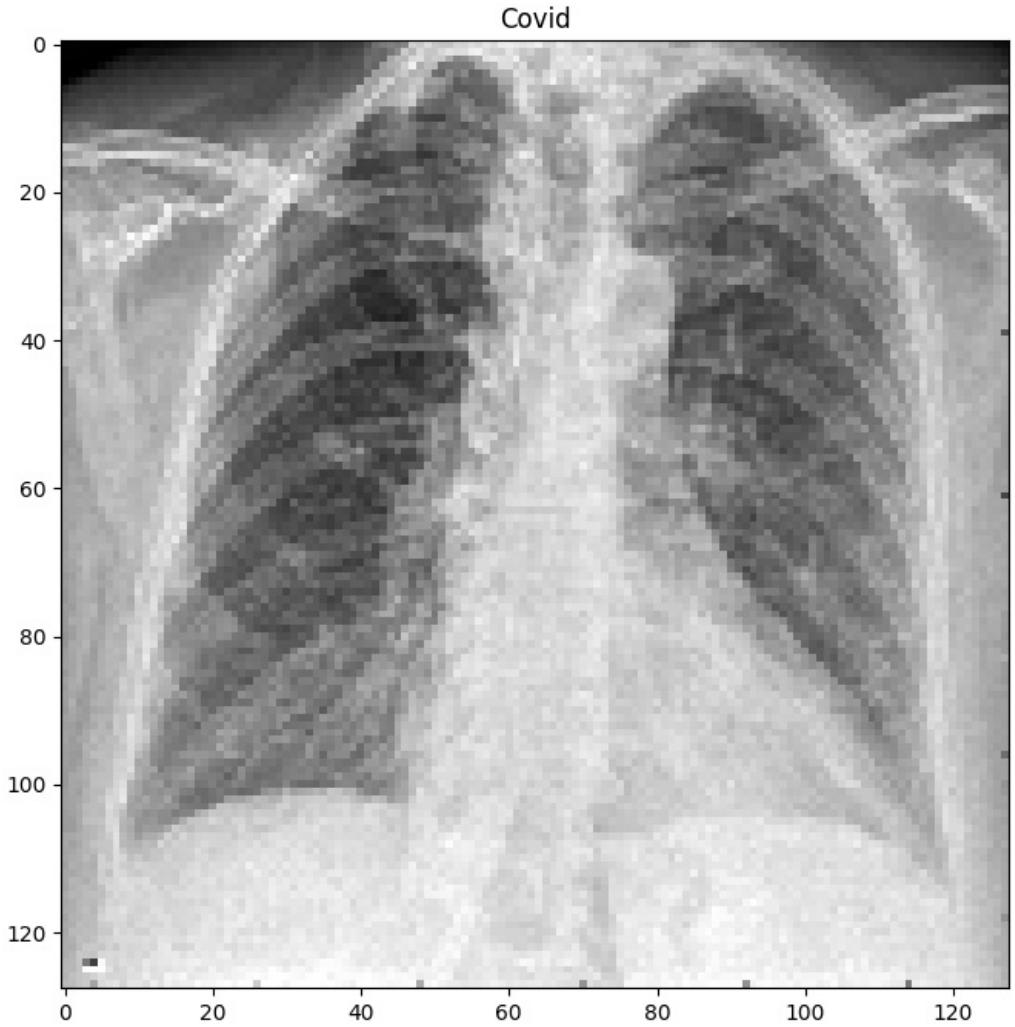

Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

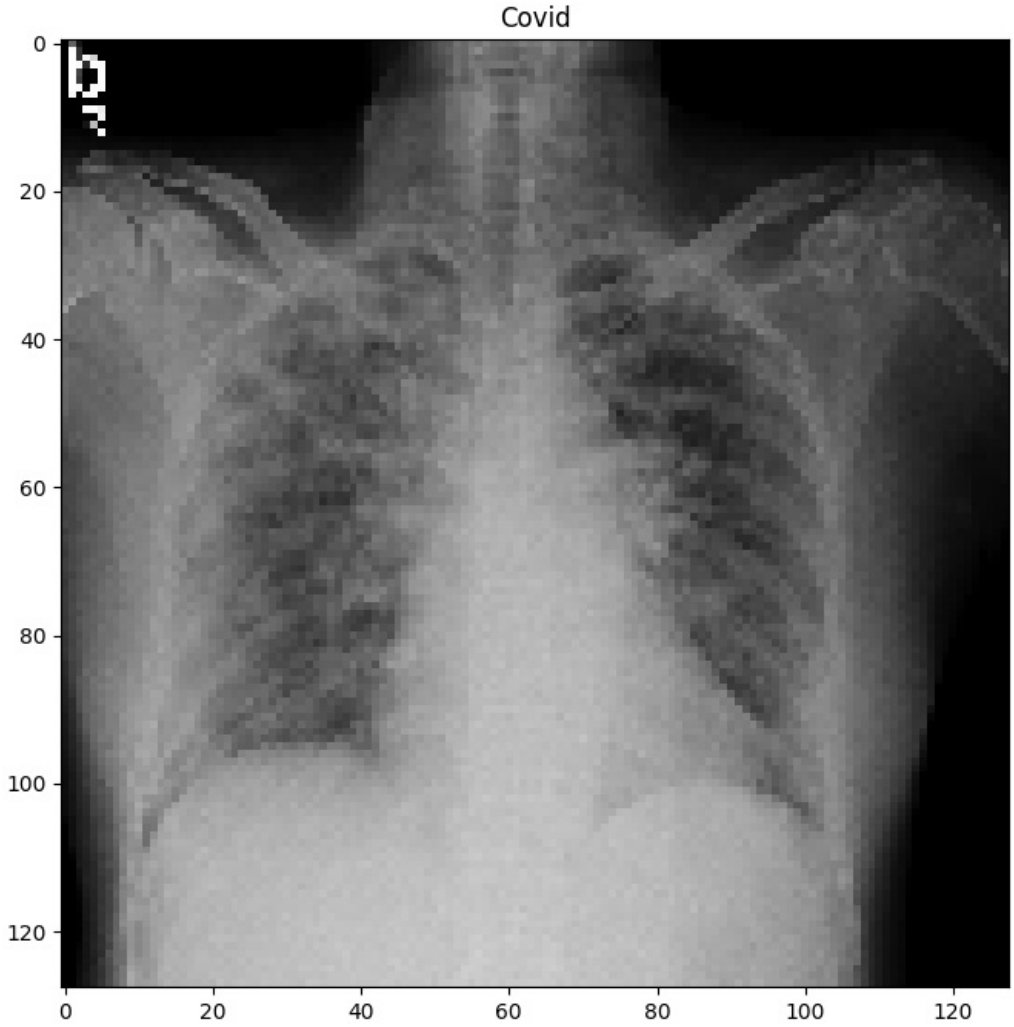

Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

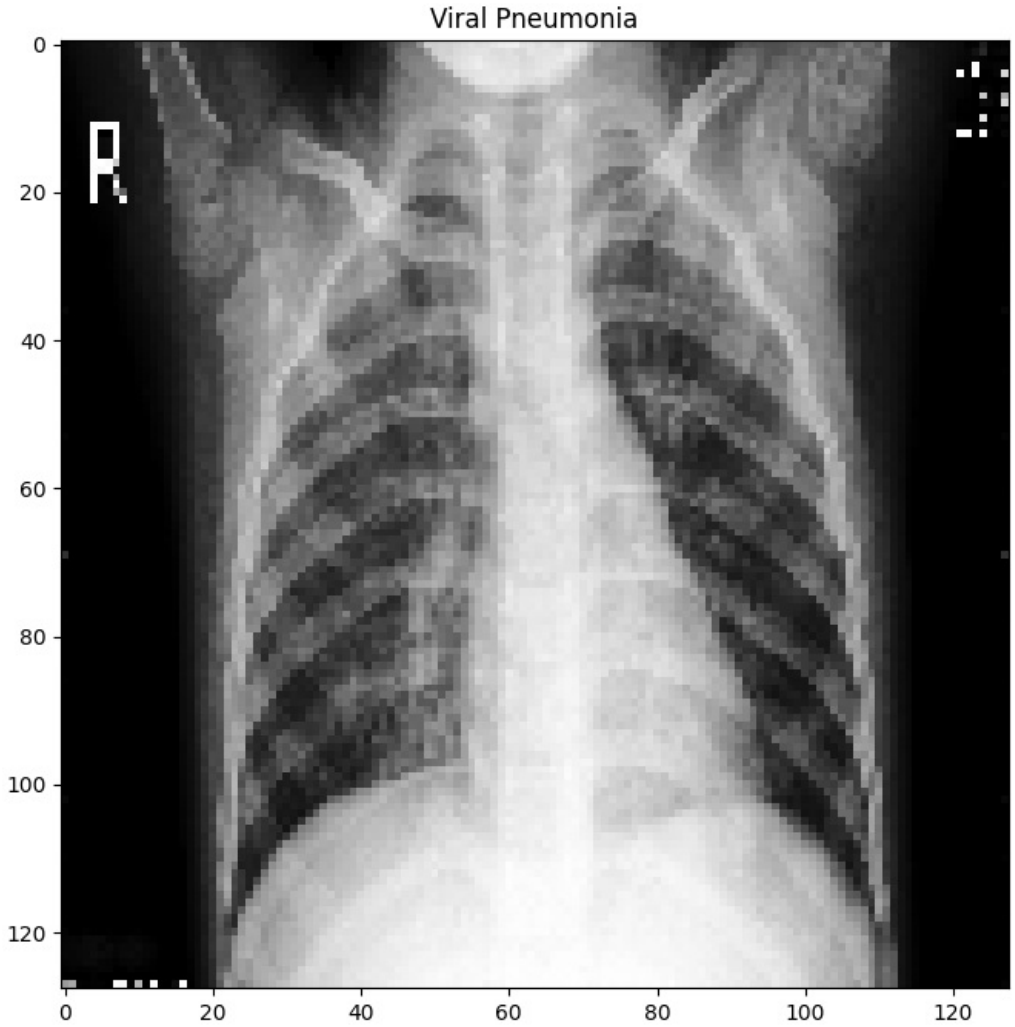

Viral Pneumonia
R
0
20
40
60
80
100
120
0
20
40
60
80
100
120

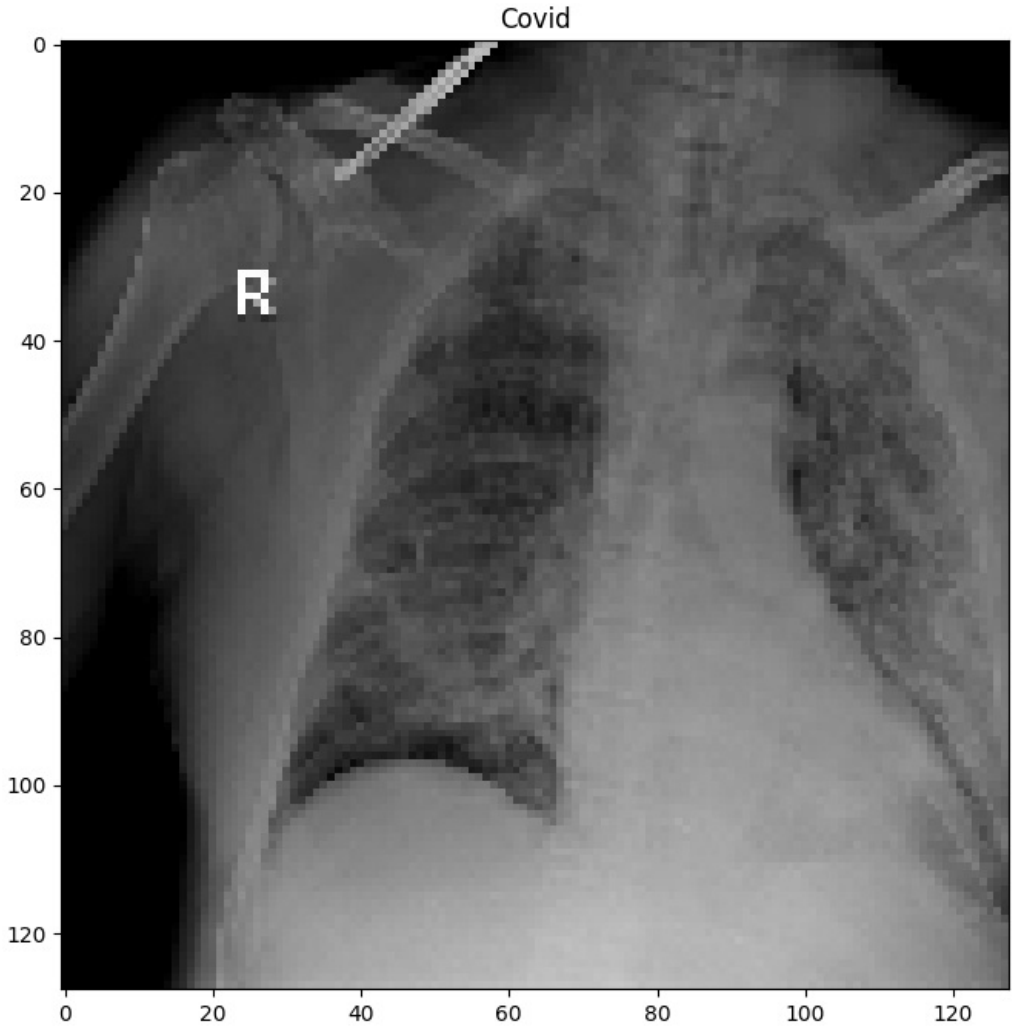

Viral Pneumonia
H
0
20
40
60
80
100
120
0
20
40
60
80
100
120


Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120


Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120


Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120


Viral Pneumonia
R
0
20
40
60
80
100
120
0
20
40
60
80
100
120



In [11]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dense,Flatten,BatchNormalization,Dropout
```

In [12]:
```
model=Sequential()
model.add(Conv2D(32,(3,3),activation="relu",input_shape=(128,128,3)))
model.add(BatchNormalization())
model.add(MaxPooling2D(3,3))
model.add(Dropout(0.5))
model.add(Conv2D(64,(3,3),activation="relu"))
model.add(BatchNormalization())
model.add(MaxPooling2D(3,3))
model.add(Dropout(0.5))
model.add(Conv2D(128,(3,3),activation="relu"))
model.add(BatchNormalization())
model.add(MaxPooling2D(3,3))
model.add(Dropout(0.5))
model.add(Conv2D(64,(3,3),activation="relu"))
model.add(Flatten())
```

```
C:\Users\vanam\Downloads\gowtham\Lib\site-packages\keras\src\layers\convolutional\base_conv.py:107: UserWarning:
Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Inp
ut(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

In [13]:
```
model.add(Dense(50,activation="relu"))
model.add(Dropout(0.6))
```

In [14]:
```
model.add(Dense(3,activation="softmax"))
```

In [15]:
```
model.compile(optimizer="adam",metrics=["accuracy"],loss="categorical_crossentropy")
```

In [16]:
```
from sklearn.preprocessing import OneHotEncoder
```

In [17]:
```
test_path=r"C:\Users\vanam\Downloads\archive covid_test_data\Covid19-dataset\test"
```

In [18]:
```
test_data=tf.keras.preprocessing.image_dataset_from_directory(test_path,image_size=(128,128))
```

```
Found 66 files belonging to 3 classes.
```

In [19]:
```
from tensorflow.keras.utils import to_categorical
train_labels = to_categorical(all_classes, num_classes=3)
```

In [20]:
```
test_images=[]
test_classes=[]
for image,classe in test_data:
    test_images.append(image.numpy())
    test_classes.append(classe)
test_images=np.concatenate(test_images,axis=0)
test_classes=np.concatenate(test_classes,axis=0)
test_images/=255
```

In [21]:
```
test_labels=to_categorical(test_classes,num_classes=3)
```

In [22]:
```
report=model.fit(all_images,train_labels,epochs=100,validation_data=(test_images,test_labels))
```

```
Epoch 1/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 14s 461ms/step - accuracy: 0.3976 - loss: 2.2606 - val_accuracy: 0.6364 - val_loss: 1.0
760
Epoch 2/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 354ms/step - accuracy: 0.4705 - loss: 1.1013 - val_accuracy: 0.3939 - val_loss: 1.03
73
Epoch 3/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 346ms/step - accuracy: 0.6044 - loss: 0.9253 - val_accuracy: 0.3939 - val_loss: 1.41
76
Epoch 4/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 419ms/step - accuracy: 0.6578 - loss: 0.8169 - val_accuracy: 0.3939 - val_loss: 2.02
40
Epoch 5/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 4s 451ms/step - accuracy: 0.6137 - loss: 0.8233 - val_accuracy: 0.3939 - val_loss: 2.75
91
Epoch 6/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 339ms/step - accuracy: 0.6679 - loss: 0.7019 - val_accuracy: 0.3939 - val_loss: 3.40
16
Epoch 7/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 335ms/step - accuracy: 0.6823 - loss: 0.6135 - val_accuracy: 0.3939 - val_loss: 4.12
86
Epoch 8/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 5s 582ms/step - accuracy: 0.7285 - loss: 0.5743 - val_accuracy: 0.3939 - val_loss: 4.89
83
Epoch 9/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 343ms/step - accuracy: 0.7071 - loss: 0.6114 - val_accuracy: 0.3939 - val_loss: 5.42
22
Epoch 10/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 5s 330ms/step - accuracy: 0.8362 - loss: 0.4372 - val_accuracy: 0.3939 - val_loss: 6.46
73
Epoch 11/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 329ms/step - accuracy: 0.7694 - loss: 0.5321 - val_accuracy: 0.3939 - val_loss: 7.65
42
Epoch 12/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 333ms/step - accuracy: 0.8361 - loss: 0.4213 - val_accuracy: 0.3939 - val_loss: 8.47
92
Epoch 13/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 384ms/step - accuracy: 0.7699 - loss: 0.4854 - val_accuracy: 0.3939 - val_loss: 9.37
59
Epoch 14/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 371ms/step - accuracy: 0.7928 - loss: 0.4232 - val_accuracy: 0.3939 - val_loss: 9.95
58
Epoch 15/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 382ms/step - accuracy: 0.8238 - loss: 0.4085 - val_accuracy: 0.3939 - val_loss: 10.4
495
Epoch 16/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 317ms/step - accuracy: 0.8101 - loss: 0.4033 - val_accuracy: 0.3939 - val_loss: 11.0
125
Epoch 17/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 353ms/step - accuracy: 0.8546 - loss: 0.3303 - val_accuracy: 0.3939 - val_loss: 11.3
579
Epoch 18/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 332ms/step - accuracy: 0.8830 - loss: 0.3228 - val_accuracy: 0.3939 - val_loss: 11.8
631
Epoch 19/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 288ms/step - accuracy: 0.8884 - loss: 0.3124 - val_accuracy: 0.3939 - val_loss: 12.0
314
Epoch 20/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 256ms/step - accuracy: 0.8291 - loss: 0.3812 - val_accuracy: 0.3939 - val_loss: 13.2
241
Epoch 21/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 246ms/step - accuracy: 0.9085 - loss: 0.3220 - val_accuracy: 0.3939 - val_loss: 13.4
382
Epoch 22/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 282ms/step - accuracy: 0.8956 - loss: 0.3056 - val_accuracy: 0.3939 - val_loss: 14.1
```

```
687
Epoch 23/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 258ms/step - accuracy: 0.8404 - loss: 0.3182 - val_accuracy: 0.3939 - val_loss: 14.8
839
Epoch 24/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 263ms/step - accuracy: 0.8797 - loss: 0.2890 - val_accuracy: 0.3939 - val_loss: 15.4
303
Epoch 25/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 249ms/step - accuracy: 0.9238 - loss: 0.2019 - val_accuracy: 0.3939 - val_loss: 15.4
033
Epoch 26/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 251ms/step - accuracy: 0.8978 - loss: 0.2896 - val_accuracy: 0.3939 - val_loss: 16.0
267
Epoch 27/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 263ms/step - accuracy: 0.9165 - loss: 0.2572 - val_accuracy: 0.3939 - val_loss: 16.7
336
Epoch 28/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 240ms/step - accuracy: 0.8666 - loss: 0.3055 - val_accuracy: 0.3939 - val_loss: 15.4
237
Epoch 29/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 236ms/step - accuracy: 0.8250 - loss: 0.3701 - val_accuracy: 0.3939 - val_loss: 15.7
081
Epoch 30/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 236ms/step - accuracy: 0.9030 - loss: 0.2150 - val_accuracy: 0.3939 - val_loss: 16.4
971
Epoch 31/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 234ms/step - accuracy: 0.9025 - loss: 0.2435 - val_accuracy: 0.3939 - val_loss: 16.8
591
Epoch 32/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 239ms/step - accuracy: 0.9435 - loss: 0.1790 - val_accuracy: 0.3939 - val_loss: 16.2
658
Epoch 33/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 247ms/step - accuracy: 0.9719 - loss: 0.1265 - val_accuracy: 0.3939 - val_loss: 16.3
397
Epoch 34/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 234ms/step - accuracy: 0.9523 - loss: 0.1943 - val_accuracy: 0.3939 - val_loss: 15.9
376
Epoch 35/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 235ms/step - accuracy: 0.9400 - loss: 0.2150 - val_accuracy: 0.3939 - val_loss: 15.7
937
Epoch 36/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 232ms/step - accuracy: 0.8939 - loss: 0.2456 - val_accuracy: 0.3939 - val_loss: 15.2
138
Epoch 37/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 236ms/step - accuracy: 0.9475 - loss: 0.1581 - val_accuracy: 0.3939 - val_loss: 14.7
805
Epoch 38/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 253ms/step - accuracy: 0.9163 - loss: 0.1786 - val_accuracy: 0.3939 - val_loss: 15.3
352
Epoch 39/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 231ms/step - accuracy: 0.9080 - loss: 0.1696 - val_accuracy: 0.3939 - val_loss: 15.8
278
Epoch 40/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 233ms/step - accuracy: 0.9123 - loss: 0.2479 - val_accuracy: 0.3939 - val_loss: 15.2
167
Epoch 41/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 233ms/step - accuracy: 0.9711 - loss: 0.1211 - val_accuracy: 0.3939 - val_loss: 13.0
033
Epoch 42/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 238ms/step - accuracy: 0.9326 - loss: 0.1743 - val_accuracy: 0.3939 - val_loss: 12.8
167
Epoch 43/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 239ms/step - accuracy: 0.8994 - loss: 0.2734 - val_accuracy: 0.3939 - val_loss: 12.3
017
Epoch 44/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 262ms/step - accuracy: 0.9650 - loss: 0.1087 - val_accuracy: 0.3939 - val_loss: 12.2
180
Epoch 45/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 234ms/step - accuracy: 0.9424 - loss: 0.1698 - val_accuracy: 0.3939 - val_loss: 13.2
328
Epoch 46/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 231ms/step - accuracy: 0.9244 - loss: 0.1832 - val_accuracy: 0.3939 - val_loss: 13.6
521
Epoch 47/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 234ms/step - accuracy: 0.9358 - loss: 0.1200 - val_accuracy: 0.3939 - val_loss: 12.1
028
Epoch 48/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 236ms/step - accuracy: 0.9691 - loss: 0.1115 - val_accuracy: 0.3939 - val_loss: 11.4
433
Epoch 49/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 239ms/step - accuracy: 0.9603 - loss: 0.1593 - val_accuracy: 0.4091 - val_loss: 10.8
276
Epoch 50/100
```

```
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 260ms/step - accuracy: 0.9483 - loss: 0.1329 - val_accuracy: 0.4242 - val_loss: 8.29
97
Epoch 51/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 256ms/step - accuracy: 0.9275 - loss: 0.1719 - val_accuracy: 0.4242 - val_loss: 8.13
31
Epoch 52/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 259ms/step - accuracy: 0.9723 - loss: 0.1298 - val_accuracy: 0.4242 - val_loss: 9.67
12
Epoch 53/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 279ms/step - accuracy: 0.9292 - loss: 0.1443 - val_accuracy: 0.4091 - val_loss: 10.6
438
Epoch 54/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 275ms/step - accuracy: 0.9564 - loss: 0.1239 - val_accuracy: 0.4242 - val_loss: 10.2
707
Epoch 55/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 291ms/step - accuracy: 0.9701 - loss: 0.1118 - val_accuracy: 0.4242 - val_loss: 10.2
137
Epoch 56/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 333ms/step - accuracy: 0.9612 - loss: 0.1475 - val_accuracy: 0.4545 - val_loss: 7.53
72
Epoch 57/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 268ms/step - accuracy: 0.9189 - loss: 0.1603 - val_accuracy: 0.4848 - val_loss: 5.71
85
Epoch 58/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 296ms/step - accuracy: 0.9656 - loss: 0.0959 - val_accuracy: 0.5000 - val_loss: 4.72
28
Epoch 59/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 357ms/step - accuracy: 0.9380 - loss: 0.1555 - val_accuracy: 0.5758 - val_loss: 2.82
97
Epoch 60/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 403ms/step - accuracy: 0.9489 - loss: 0.1367 - val_accuracy: 0.5909 - val_loss: 2.66
95
Epoch 61/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 4s 499ms/step - accuracy: 0.9351 - loss: 0.2040 - val_accuracy: 0.5152 - val_loss: 5.03
14
Epoch 62/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 359ms/step - accuracy: 0.9335 - loss: 0.1506 - val_accuracy: 0.5152 - val_loss: 5.26
49
Epoch 63/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 351ms/step - accuracy: 0.9591 - loss: 0.1105 - val_accuracy: 0.4697 - val_loss: 6.48
54
Epoch 64/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 343ms/step - accuracy: 0.9543 - loss: 0.1346 - val_accuracy: 0.4394 - val_loss: 9.11
58
Epoch 65/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 330ms/step - accuracy: 0.9612 - loss: 0.1614 - val_accuracy: 0.4394 - val_loss: 9.07
80
Epoch 66/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 343ms/step - accuracy: 0.9779 - loss: 0.1202 - val_accuracy: 0.4394 - val_loss: 8.44
96
Epoch 67/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 339ms/step - accuracy: 0.9701 - loss: 0.0976 - val_accuracy: 0.4394 - val_loss: 8.87
07
Epoch 68/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 354ms/step - accuracy: 0.9740 - loss: 0.0828 - val_accuracy: 0.4697 - val_loss: 7.09
89
Epoch 69/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 383ms/step - accuracy: 0.9790 - loss: 0.0825 - val_accuracy: 0.5606 - val_loss: 4.28
64
Epoch 70/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 354ms/step - accuracy: 0.9647 - loss: 0.1047 - val_accuracy: 0.6212 - val_loss: 3.25
62
Epoch 71/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 428ms/step - accuracy: 0.9536 - loss: 0.1337 - val_accuracy: 0.6061 - val_loss: 3.49
36
Epoch 72/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 337ms/step - accuracy: 0.9772 - loss: 0.0715 - val_accuracy: 0.7424 - val_loss: 2.01
44
Epoch 73/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 337ms/step - accuracy: 0.9548 - loss: 0.0885 - val_accuracy: 0.5606 - val_loss: 4.85
02
Epoch 74/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 431ms/step - accuracy: 0.9783 - loss: 0.0826 - val_accuracy: 0.6667 - val_loss: 2.85
26
Epoch 75/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 337ms/step - accuracy: 0.9768 - loss: 0.0685 - val_accuracy: 0.8030 - val_loss: 1.41
08
Epoch 76/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 320ms/step - accuracy: 0.9972 - loss: 0.0340 - val_accuracy: 0.7727 - val_loss: 1.89
35
Epoch 77/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 303ms/step - accuracy: 0.9689 - loss: 0.0834 - val_accuracy: 0.7576 - val_loss: 2.18
71
```

```
Epoch 78/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 311ms/step - accuracy: 0.9695 - loss: 0.0577 - val_accuracy: 0.7879 - val_loss: 1.50
46
Epoch 79/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 331ms/step - accuracy: 0.9714 - loss: 0.1141 - val_accuracy: 0.7879 - val_loss: 1.43
87
Epoch 80/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 416ms/step - accuracy: 0.9688 - loss: 0.1108 - val_accuracy: 0.8485 - val_loss: 0.98
87
Epoch 81/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 343ms/step - accuracy: 0.9789 - loss: 0.0547 - val_accuracy: 0.8485 - val_loss: 0.69
16
Epoch 82/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 4s 477ms/step - accuracy: 0.9602 - loss: 0.0713 - val_accuracy: 0.7879 - val_loss: 1.36
02
Epoch 83/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 362ms/step - accuracy: 0.9809 - loss: 0.0552 - val_accuracy: 0.7879 - val_loss: 1.60
05
Epoch 84/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 333ms/step - accuracy: 0.9543 - loss: 0.1190 - val_accuracy: 0.7879 - val_loss: 1.39
69
Epoch 85/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 337ms/step - accuracy: 0.9763 - loss: 0.0732 - val_accuracy: 0.7879 - val_loss: 1.83
81
Epoch 86/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 332ms/step - accuracy: 0.9732 - loss: 0.0875 - val_accuracy: 0.8030 - val_loss: 1.46
24
Epoch 87/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 338ms/step - accuracy: 0.9621 - loss: 0.0854 - val_accuracy: 0.8182 - val_loss: 1.38
16
Epoch 88/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 387ms/step - accuracy: 0.9825 - loss: 0.0698 - val_accuracy: 0.7879 - val_loss: 1.55
86
Epoch 89/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 369ms/step - accuracy: 0.9771 - loss: 0.0676 - val_accuracy: 0.7727 - val_loss: 1.70
03
Epoch 90/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 341ms/step - accuracy: 0.9759 - loss: 0.0670 - val_accuracy: 0.8030 - val_loss: 1.32
97
Epoch 91/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 337ms/step - accuracy: 0.9609 - loss: 0.0977 - val_accuracy: 0.8333 - val_loss: 1.21
73
Epoch 92/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 317ms/step - accuracy: 0.9563 - loss: 0.0995 - val_accuracy: 0.9242 - val_loss: 0.48
31
Epoch 93/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 321ms/step - accuracy: 0.9916 - loss: 0.0477 - val_accuracy: 0.9394 - val_loss: 0.22
68
Epoch 94/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 330ms/step - accuracy: 0.9900 - loss: 0.0537 - val_accuracy: 0.9394 - val_loss: 0.15
67
Epoch 95/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 334ms/step - accuracy: 0.9907 - loss: 0.0465 - val_accuracy: 0.9242 - val_loss: 0.15
42
Epoch 96/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 318ms/step - accuracy: 0.9682 - loss: 0.0845 - val_accuracy: 0.9091 - val_loss: 0.25
25
Epoch 97/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 318ms/step - accuracy: 0.9494 - loss: 0.1444 - val_accuracy: 0.9394 - val_loss: 0.35
33
Epoch 98/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 322ms/step - accuracy: 0.9628 - loss: 0.0690 - val_accuracy: 0.8636 - val_loss: 0.73
02
Epoch 99/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 2s 309ms/step - accuracy: 0.9648 - loss: 0.0773 - val_accuracy: 0.8636 - val_loss: 1.33
40
Epoch 100/100
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 309ms/step - accuracy: 0.9763 - loss: 0.0719 - val_accuracy: 0.9242 - val_loss: 0.40
20
```

In [23]:

```
from collections import Counter
print(Counter(all_classes))
```

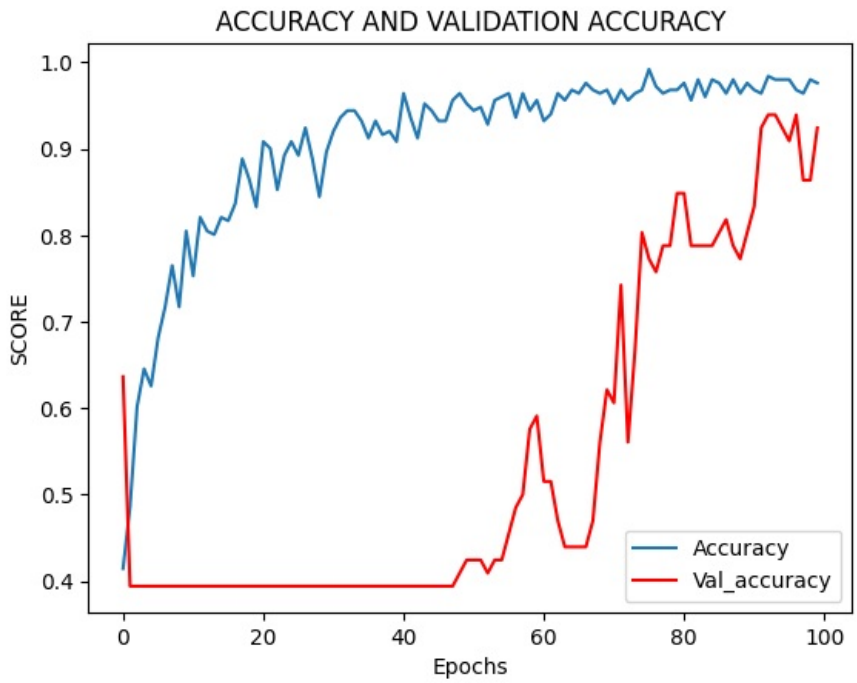
```
Counter({0: 111, 1: 70, 2: 70})
```

In [24]:

```
plt.title("ACCURACY AND VALIDATION ACCURACY")
plt.plot(report.history["accuracy"],label="Accuracy")
plt.plot(report.history["val_accuracy"],color="red",label="Val_accuracy")
plt.xlabel("Epochs")
plt.ylabel("SCORE")
plt.legend(loc="lower right")
```

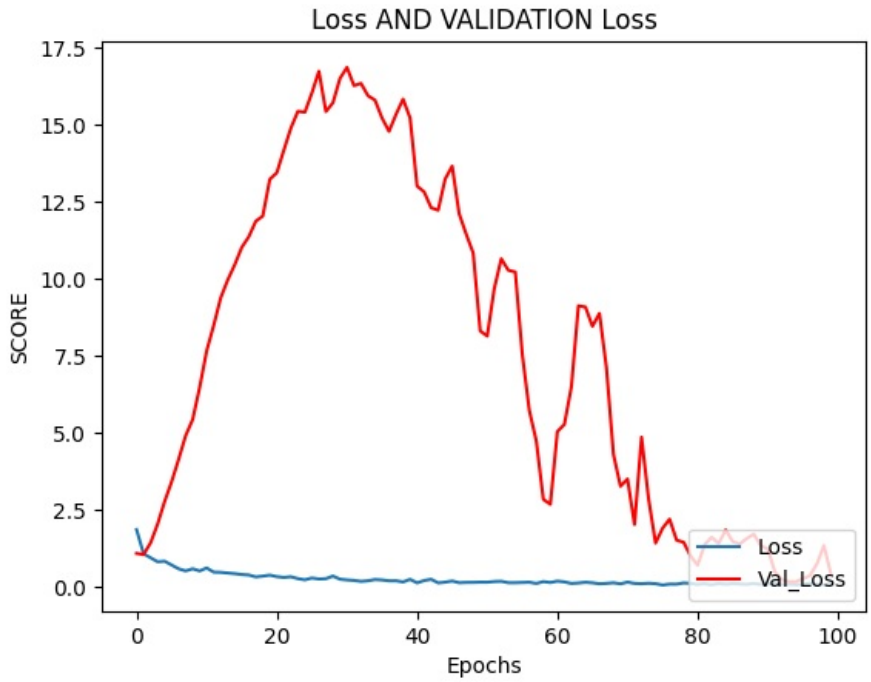
Out[24]:

```
<matplotlib.legend.Legend at 0x1e55c5668a0>
```



In [34]:
```
plt.title("Loss AND VALIDATION Loss")
plt.plot(report.history["loss"],label="Loss")
plt.plot(report.history["val_loss"],color="red",label="Val_Loss")
plt.xlabel("Epochs")
plt.ylabel("SCORE")
plt.legend(loc="lower right")
```

Out[34]:
```
<matplotlib.legend.Legend at 0x1e55c41fd70>
```



In [25]:
```
test_images
```

Out[25]:
```
array([[[[8.10260847e-02, 8.10260847e-02, 8.10260847e-02],
         [8.89399499e-02, 8.89399499e-02, 8.89399499e-02],
         [9.61357579e-02, 9.61357579e-02, 9.61357579e-02],
         ...,
         [7.45098069e-02, 7.45098069e-02, 7.45098069e-02],
         [7.05882385e-02, 7.05882385e-02, 7.05882385e-02],
         [7.83029571e-02, 7.83029571e-02, 7.83029571e-02]],

        [[9.58180130e-02, 9.58180130e-02, 9.58180130e-02],
         [8.95850062e-02, 8.95850062e-02, 8.95850062e-02],
         [8.53208080e-02, 8.53208080e-02, 8.53208080e-02],
         ...,
         [7.69568384e-02, 7.69568384e-02, 7.69568384e-02],
         [7.60195255e-02, 7.60195255e-02, 7.60195255e-02],
         [6.92731068e-02, 6.92731068e-02, 6.92731068e-02]],
```

```
 [[8.84191170e-02, 8.84191170e-02, 8.84191170e-02],
  [9.16053951e-02, 9.16053951e-02, 9.16053951e-02],
  [8.81237537e-02, 8.81237537e-02, 8.81237537e-02],
  ...,
  [7.39430115e-02, 7.39430115e-02, 7.39430115e-02],
  [6.93321079e-02, 6.93321079e-02, 6.93321079e-02],
  [6.36335760e-02, 6.36335760e-02, 6.36335760e-02]],

 ...,

 [[0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  ...,
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]],

 [[0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  ...,
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]],

 [[0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  ...,
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]]],


[[[2.27450982e-01, 2.19607845e-01, 2.31372550e-01],
  [2.27450982e-01, 2.19607845e-01, 2.31372550e-01],
  [2.28429452e-01, 2.20586315e-01, 2.32351020e-01],
  ...,
  [2.27450982e-01, 2.19607845e-01, 2.31372550e-01],
  [2.27450982e-01, 2.19607845e-01, 2.31372550e-01],
  [2.27450982e-01, 2.19607845e-01, 2.31372550e-01]],

 [[2.27450982e-01, 2.19607845e-01, 2.31372550e-01],
  [2.27450982e-01, 2.19607845e-01, 2.31372550e-01],
  [2.26332724e-01, 2.18489587e-01, 2.30254292e-01],
  ...,
  [2.27450982e-01, 2.19607845e-01, 2.31372550e-01],
  [2.27450982e-01, 2.19607845e-01, 2.31372550e-01],
  [2.27450982e-01, 2.19607845e-01, 2.31372550e-01]],

 [[2.27450982e-01, 2.19607845e-01, 2.28921562e-01],
  [2.26996213e-01, 2.20062613e-01, 2.30462998e-01],
  [2.16943353e-01, 2.10852295e-01, 2.19289988e-01],
  ...,
  [2.27450982e-01, 2.19607845e-01, 2.31372550e-01],
  [2.27450982e-01, 2.19607845e-01, 2.31372550e-01],
  [2.27450982e-01, 2.19607845e-01, 2.31372550e-01]],

 ...,

 [[2.24993303e-01, 2.17150167e-01, 2.28914872e-01],
  [2.23074645e-01, 2.21073642e-01, 2.26526111e-01],
  [2.42545575e-01, 2.46749580e-01, 2.43418738e-01],
  ...,
  [2.27450982e-01, 2.19607845e-01, 2.31372550e-01],
  [2.27450982e-01, 2.19607845e-01, 2.31372550e-01],
  [2.27450982e-01, 2.19607845e-01, 2.31372550e-01]],

 [[2.25658700e-01, 2.17815563e-01, 2.26667821e-01],
  [2.23529413e-01, 2.20618874e-01, 2.28446692e-01],
  [2.36518651e-01, 2.42125273e-01, 2.34766588e-01],
  ...,
  [2.27450982e-01, 2.19607845e-01, 2.31372550e-01],
  [2.27450982e-01, 2.19607845e-01, 2.31372550e-01],
  [2.27450982e-01, 2.19607845e-01, 2.31372550e-01]],

 [[2.27338970e-01, 2.19495833e-01, 2.27044076e-01],
  [2.24893734e-01, 2.21983194e-01, 2.23928660e-01],
  [2.39607275e-01, 2.42585778e-01, 2.49763519e-01],
  ...,
  [2.27450982e-01, 2.19607845e-01, 2.31372550e-01],
  [2.27450982e-01, 2.19607845e-01, 2.31372550e-01],
```

```
  [2.27450982e-01, 2.19607845e-01, 2.31372550e-01]]],


[[[7.05523342e-02, 7.05523342e-02, 7.05523342e-02],
  [8.53395984e-02, 8.53395984e-02, 8.53395984e-02],
  [1.28855988e-01, 1.28855988e-01, 1.28855988e-01],
  ...,
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]],

 [[6.61764741e-02, 6.61764741e-02, 6.61764741e-02],
  [9.11735967e-02, 9.11735967e-02, 9.11735967e-02],
  [1.25214458e-01, 1.25214458e-01, 1.25214458e-01],
  ...,
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]],

 [[6.88376054e-02, 6.88376054e-02, 6.88376054e-02],
  [7.79765993e-02, 7.79765993e-02, 7.79765993e-02],
  [1.05615713e-01, 1.05615713e-01, 1.05615713e-01],
  ...,
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]],

 ...,

 [[0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  ...,
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]],

 [[0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  ...,
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]],

 [[0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  ...,
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]]],


...,


[[[5.84726371e-02, 5.84726371e-02, 5.84726371e-02],
  [6.82598054e-02, 6.82598054e-02, 6.82598054e-02],
  [7.92657584e-02, 7.92657584e-02, 7.92657584e-02],
  ...,
  [1.22403011e-01, 1.22403011e-01, 1.22403011e-01],
  [1.24433212e-01, 1.24433212e-01, 1.24433212e-01],
  [1.28169522e-01, 1.28169522e-01, 1.28169522e-01]],

 [[6.09638281e-02, 6.09638281e-02, 6.09638281e-02],
  [7.07390308e-02, 7.07390308e-02, 7.07390308e-02],
  [7.97794089e-02, 7.97794089e-02, 7.97794089e-02],
  ...,
  [1.34466916e-01, 1.34466916e-01, 1.34466916e-01],
  [1.46798402e-01, 1.46798402e-01, 1.46798402e-01],
  [1.15563728e-01, 1.15563728e-01, 1.15563728e-01]],

 [[5.87244369e-02, 5.87244369e-02, 5.87244369e-02],
  [6.93598762e-02, 6.93598762e-02, 6.93598762e-02],
  [8.97724256e-02, 8.97724256e-02, 8.97724256e-02],
  ...,
  [1.17136277e-01, 1.17136277e-01, 1.17136277e-01],
  [1.23400636e-01, 1.23400636e-01, 1.23400636e-01],
  [1.29755959e-01, 1.29755959e-01, 1.29755959e-01]],

 ...,
```

```
 [[0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  ...,
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]],

 [[0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  ...,
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]],

 [[0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  ...,
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]]],


[[[0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  ...,
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]],

 [[0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  ...,
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]],

 [[0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  ...,
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]],

 ...,

 [[0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  ...,
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]],

 [[0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  ...,
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]],

 [[0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  ...,
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
  [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]]],


[[[8.60906839e-02, 8.60906839e-02, 8.60906839e-02],
  [8.48879367e-02, 8.48879367e-02, 8.48879367e-02],
  [1.19318940e-01, 1.19318940e-01, 1.19318940e-01],
  ...,
  [4.74365354e-01, 4.74365354e-01, 4.74365354e-01],
  [3.61588299e-01, 3.61588299e-01, 3.61588299e-01],
  [1.77697390e-01, 1.77697390e-01, 1.77697390e-01]],
```

```
       [[8.23568925e-02, 8.23568925e-02, 8.23568925e-02],
        [7.05763847e-02, 7.05763847e-02, 7.05763847e-02],
        [1.23161763e-01, 1.23161763e-01, 1.23161763e-01],
        ...,
        [4.44813460e-01, 4.44813460e-01, 4.44813460e-01],
        [3.66179943e-01, 3.66179943e-01, 3.66179943e-01],
        [2.03067675e-01, 2.03067675e-01, 2.03067675e-01]],

       [[7.53282756e-02, 7.53282756e-02, 7.53282756e-02],
        [9.83871147e-02, 9.83871147e-02, 9.83871147e-02],
        [1.15224443e-01, 1.15224443e-01, 1.15224443e-01],
        ...,
        [4.73905563e-01, 4.73905563e-01, 4.73905563e-01],
        [2.92754292e-01, 2.92754292e-01, 2.92754292e-01],
        [1.85020536e-01, 1.85020536e-01, 1.85020536e-01]],

       ...,

       [[9.19117665e-05, 9.19117665e-05, 9.19117665e-05],
        [3.31539363e-02, 3.31539363e-02, 3.31539363e-02],
        [3.54291499e-01, 3.54291499e-01, 3.54291499e-01],
        ...,
        [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
        [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
        [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]],

       [[1.68504906e-04, 1.68504906e-04, 1.68504906e-04],
        [4.34470996e-02, 4.34470996e-02, 4.34470996e-02],
        [3.26803535e-01, 3.26803535e-01, 3.26803535e-01],
        ...,
        [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
        [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
        [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]],

       [[9.19117665e-05, 9.19117665e-05, 9.19117665e-05],
        [5.81915155e-02, 5.81915155e-02, 5.81915155e-02],
        [3.37699145e-01, 3.37699145e-01, 3.37699145e-01],
        ...,
        [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
        [0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
        [0.00000000e+00, 0.00000000e+00, 0.00000000e+00]]]],
      dtype=float32)
```

In [26]:
```
predicted=model.predict(test_images)
```

```
3/3 ━━━━━━━━━━━━━━━━━━━━ 1s 345ms/step
```

In [27]:
```
predicted.shape
```

Out[27]:
```
(66, 3)
```

In [28]:
```
predicted
```

Out[28]:
```
array([[1.5435372e-06, 2.1061741e-03, 9.9789232e-01],
       [1.0000000e+00, 1.2246756e-28, 5.0195900e-26],
       [8.3793839e-04, 8.8196361e-01, 1.1719852e-01],
       [1.2420650e-04, 5.9280251e-06, 9.9986982e-01],
       [4.3353179e-05, 9.9875402e-01, 1.2025691e-03],
       [1.0000000e+00, 7.9345731e-12, 1.5756936e-10],
       [1.0000000e+00, 9.3194029e-21, 8.7672098e-19],
       [1.0000000e+00, 6.8325249e-33, 2.5429939e-30],
       [1.3986696e-06, 3.4849902e-06, 9.9999511e-01],
       [1.7238781e-04, 6.7013993e-05, 9.9976069e-01],
       [4.8421070e-02, 9.4391376e-01, 7.6651429e-03],
       [5.2869394e-03, 8.3811296e-04, 9.9387497e-01],
       [9.9999988e-01, 2.9826076e-08, 7.2896640e-08],
       [9.9996817e-01, 2.0909660e-05, 1.0989641e-05],
       [9.0484634e-02, 4.7884605e-04, 9.0903652e-01],
       [1.0000000e+00, 7.0646708e-27, 5.2070810e-25],
       [1.0000000e+00, 4.3781049e-15, 8.4626696e-14],
       [3.4558217e-04, 9.7453904e-01, 2.5115389e-02],
       [1.0000000e+00, 8.1328276e-16, 1.5248808e-13],
       [7.1115774e-04, 7.2485334e-01, 2.7443549e-01],
       [2.7956221e-05, 9.9665451e-01, 3.3174702e-03],
       [1.0000000e+00, 1.6448585e-22, 7.4572780e-21],
       [9.9973306e-07, 5.1650556e-04, 9.9948251e-01],
       [5.9688489e-07, 2.6882690e-04, 9.9973053e-01],
       [7.2029675e-04, 9.9097168e-01, 8.3080595e-03],
       [9.9975091e-01, 1.9851762e-04, 5.0488557e-05],
       [6.8480549e-05, 9.9517077e-01, 4.7607240e-03],
       [9.9998009e-01, 9.8414910e-07, 1.8927802e-05],
       [1.0000000e+00, 2.5140645e-13, 9.3767061e-10],
       [1.7692992e-05, 3.9731557e-03, 9.9600911e-01],
       [1.5586045e-01, 3.8550596e-04, 8.4375399e-01],
       [1.0000000e+00, 9.1275699e-16, 2.3778969e-12],
       [1.5995880e-04, 9.8698062e-01, 1.2859376e-02],
       [5.7706377e-04, 9.7098494e-01, 2.8437970e-02],
       [1.0000000e+00, 7.3325787e-33, 2.7489817e-30],
       [1.0000000e+00, 5.7363264e-22, 8.6560115e-20],
       [8.5963501e-04, 9.9495572e-01, 4.1846093e-03],
       [5.2376521e-01, 7.3048583e-04, 4.7550440e-01],
       [1.2917422e-04, 2.9939465e-05, 9.9984086e-01],
       [1.0000000e+00, 7.8101198e-22, 1.9608983e-19],
       [8.1978060e-06, 9.9969029e-01, 3.0155256e-04],
       [1.0000000e+00, 1.7881992e-19, 3.7803238e-18],
       [8.2285011e-01, 1.7336805e-01, 3.7817813e-03],
       [1.0000000e+00, 1.0161010e-10, 3.0369721e-11],
       [1.0000000e+00, 5.1725286e-12, 1.5488327e-12],
       [6.0165192e-05, 9.9313110e-01, 6.8087205e-03],
       [2.0734721e-03, 9.7209734e-01, 2.5829151e-02],
       [8.4577790e-05, 1.8405171e-04, 9.9973136e-01],
       [1.0000000e+00, 4.9489028e-17, 1.6837626e-14],
       [1.0000000e+00, 6.2032373e-19, 4.1368411e-17],
       [4.3660030e-03, 1.9096297e-01, 8.0467105e-01],
       [4.2876678e-08, 1.7801518e-04, 9.9982196e-01],
       [3.6857437e-02, 1.6374490e-05, 9.6312612e-01],
       [1.0000000e+00, 1.2246756e-28, 5.0195900e-26],
       [1.0000000e+00, 5.6732520e-25, 7.7571754e-23],
       [1.0000000e+00, 1.8816556e-16, 2.3600096e-15],
       [7.2729476e-02, 9.1349828e-01, 1.3772210e-02],
       [3.2325904e-04, 9.1779011e-01, 8.1886679e-02],
       [1.0000000e+00, 5.1185963e-17, 5.5916503e-17],
       [1.0000000e+00, 1.5963579e-21, 2.4017181e-19],
       [2.4818477e-05, 9.9344105e-01, 6.5341410e-03],
       [1.0000000e+00, 1.1179932e-28, 6.5721955e-28],
       [1.5926069e-02, 3.9849043e-02, 9.4422489e-01],
       [4.2538463e-06, 9.9249638e-04, 9.9900323e-01],
       [2.8484905e-05, 9.9873596e-01, 1.2355793e-03],
       [1.0525084e-03, 9.9355423e-01, 5.3932616e-03]], dtype=float32)
```

In [37]:
```
train_predicted=model.predict(all_images)
train_predicted=np.argmax(train_predicted,axis=1)
```

```
8/8 ━━━━━━━━━━━━━━━━━━━━ 1s 45ms/step
```

In [29]:
```
predicted=np.argmax(predicted,axis=1)
```

In [39]:
```
train_predicted
```

Out[39]:
```
array([1, 0, 1, 2, 0, 0, 0, 1, 0, 2, 1, 0, 0, 1, 0, 0, 0, 0, 0, 2, 2, 1,
       0, 2, 2, 0, 0, 0, 2, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 2, 0,
       2, 1, 0, 0, 0, 0, 0, 1, 2, 1, 2, 1, 2, 0, 1, 2, 2, 2, 2, 0, 1, 0,
       0, 0, 0, 0, 2, 0, 0, 1, 2, 0, 0, 0, 1, 0, 2, 0, 1, 2, 0, 2, 0, 1,
       1, 0, 0, 0, 1, 2, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0,
       0, 2, 2, 0, 0, 0, 0, 0, 2, 0, 2, 1, 0, 2, 0, 1, 0, 1, 2, 2, 1, 0,
       2, 0, 1, 0, 2, 2, 0, 2, 2, 0, 0, 0, 0, 0, 0, 0, 2, 1, 1, 0, 0, 1,
       0, 0, 1, 2, 0, 1, 0, 0, 0, 2, 1, 1, 1, 2, 2, 0, 1, 0, 0, 1, 2, 0,
       1, 0, 2, 0, 1, 1, 2, 2, 0, 0, 1, 0, 0, 2, 2, 2, 1, 2, 2, 0, 2, 1,
       2, 1, 0, 1, 0, 2, 2, 2, 2, 1, 0, 1, 1, 2, 2, 0, 0, 1, 0, 0, 2, 1,
       1, 2, 0, 0, 1, 2, 0, 1, 0, 0, 1, 0, 2, 0, 0, 0, 0, 1, 0, 2, 0, 2,
       0, 0, 2, 1, 0, 0, 2, 0, 0], dtype=int64)
```

In [31]:
```
from sklearn.metrics import confusion_matrix
```

In [32]:
```
cm1=confusion_matrix(test_classes,predicted)
```

In [42]:
```
cm_train=confusion_matrix(all_classes,train_predicted)
cm_train
```

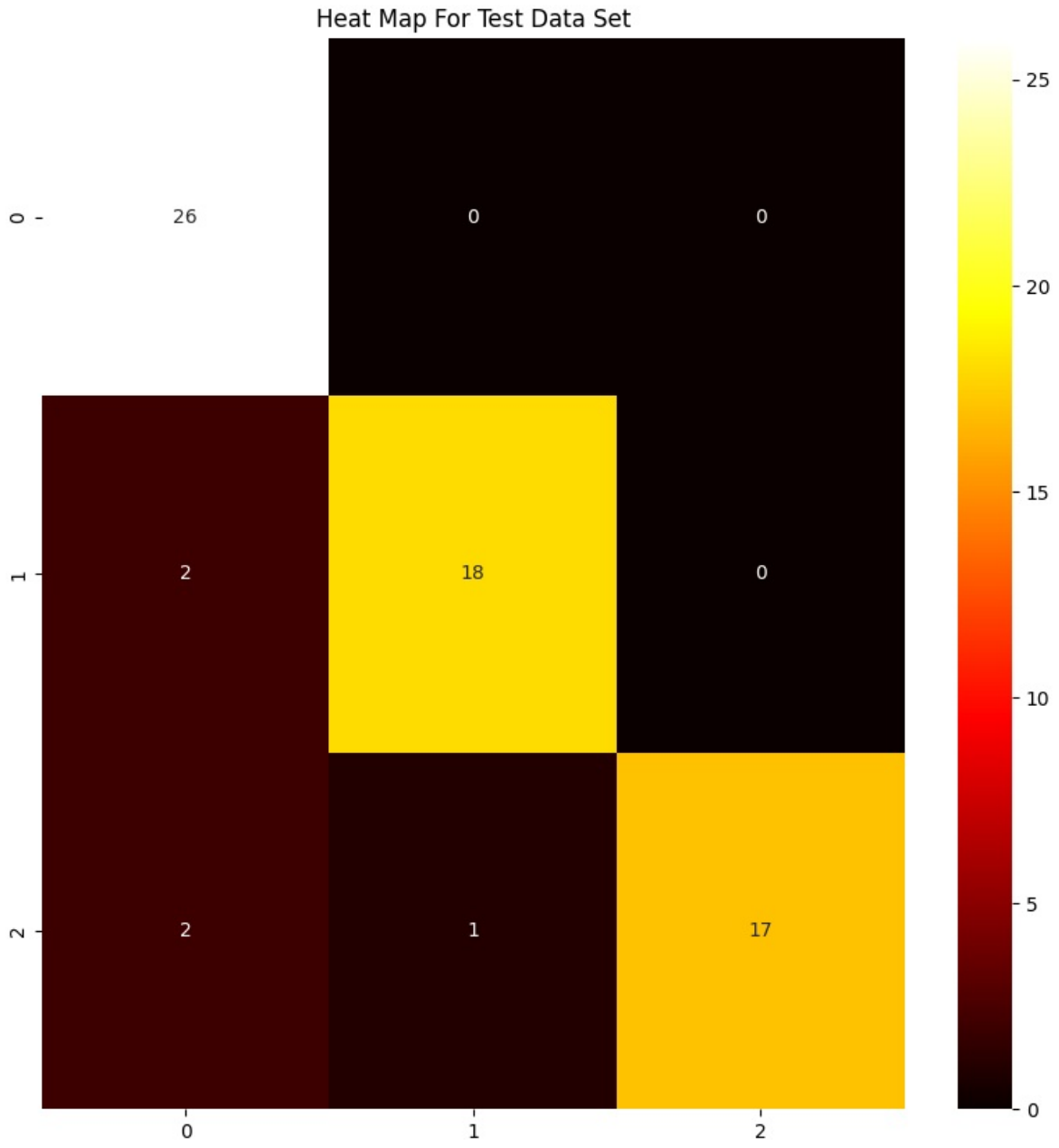
Out[42]:
```
array([[111,   0,   0],
       [ 10,  60,   0],
       [  6,   0,  64]], dtype=int64)
```

In [33]:
```
cm1
```

Out[33]:
```
array([[26,  0,  0],
       [ 2, 18,  0],
       [ 2,  1, 17]], dtype=int64)
```

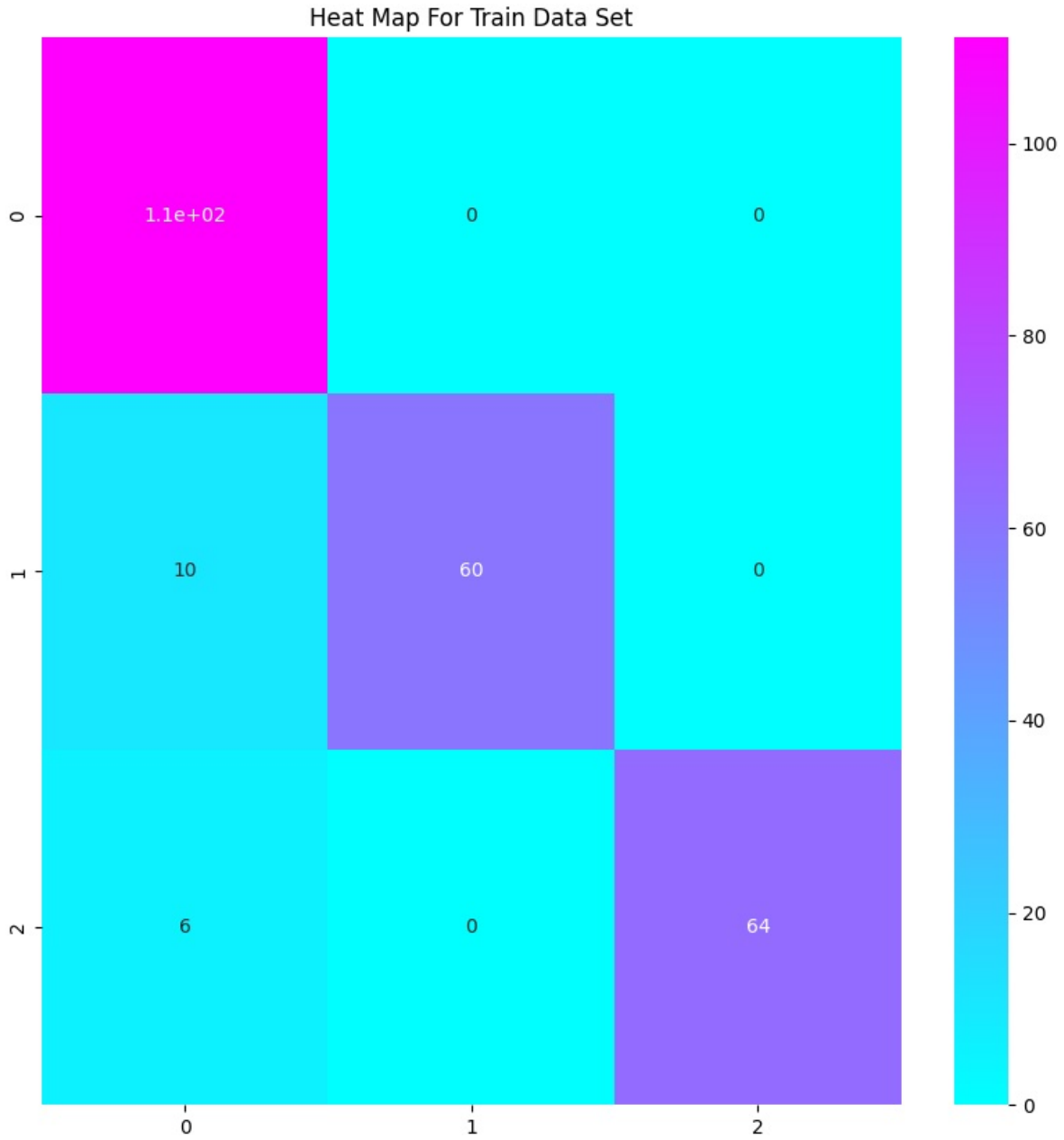
In [36]:
```
import seaborn as sea
plt.figure(figsize=(10,10))
plt.title("Heat Map For Test Data Set")
sea.heatmap(cm1,annot=True,cmap="hot")
```

Out[36]:
```
<Axes: title={'center': 'Heat Map For Test Data Set'}>
```



In [43]:
```
plt.figure(figsize=(10,10))
plt.title("Heat Map For Train Data Set")
sea.heatmap(cm_train,annot=True,cmap="cool")
```

Out[43]: <Axes: title={'center': 'Heat Map For Train Data Set'}>



In [53]:
```
from sklearn.metrics import confusion_matrix,accuracy_score,mean_squared_error,mean_absolute_error,f1_score,roc_
```

In [54]:
```
train_accuracy=accuracy_score(all_classes,train_predicted)
test_accuracy=accuracy_score(test_classes,predicted)
train_mse=mean_squared_error(all_classes,train_predicted)
```

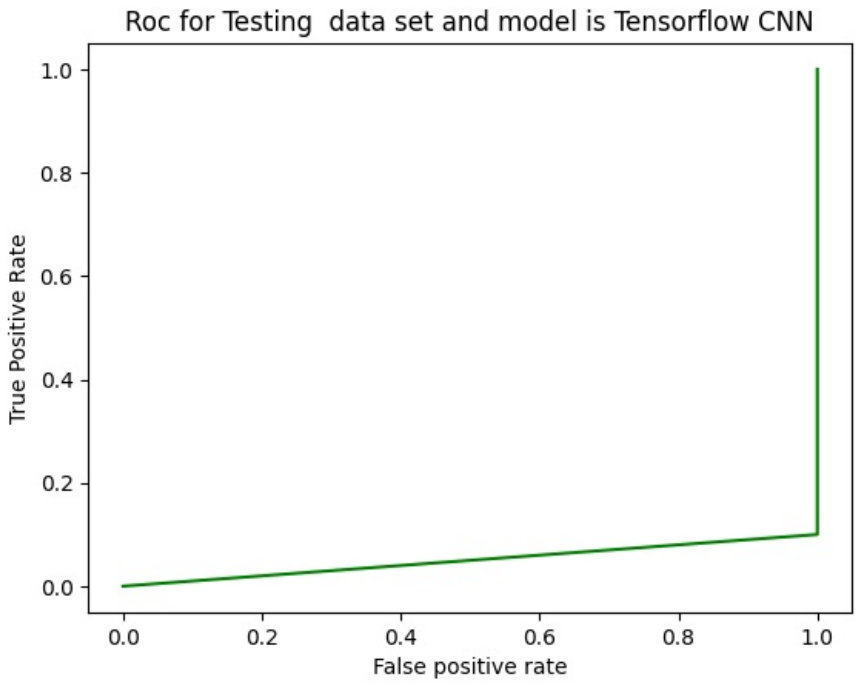
```
train_mae=mean_absolute_error(all_classes,train_predicted)
test_mse=mean_squared_error(test_classes,predicted)
test_mae=mean_absolute_error(test_classes,predicted)
```

In [55]:
```
print("train Accuracy Score = {:.2f} and Test Accuracy Score = {:.2f} train mse={}, train mae= {},test_mse={},te
```

```
train Accuracy Score = 93.63 and Test Accuracy Score = 92.42 train mse=0.13545816733067728, train mae= 0.0876494
0239043825,test_mse=0.16666666666666666,test_mae=0.10606060606060606
```

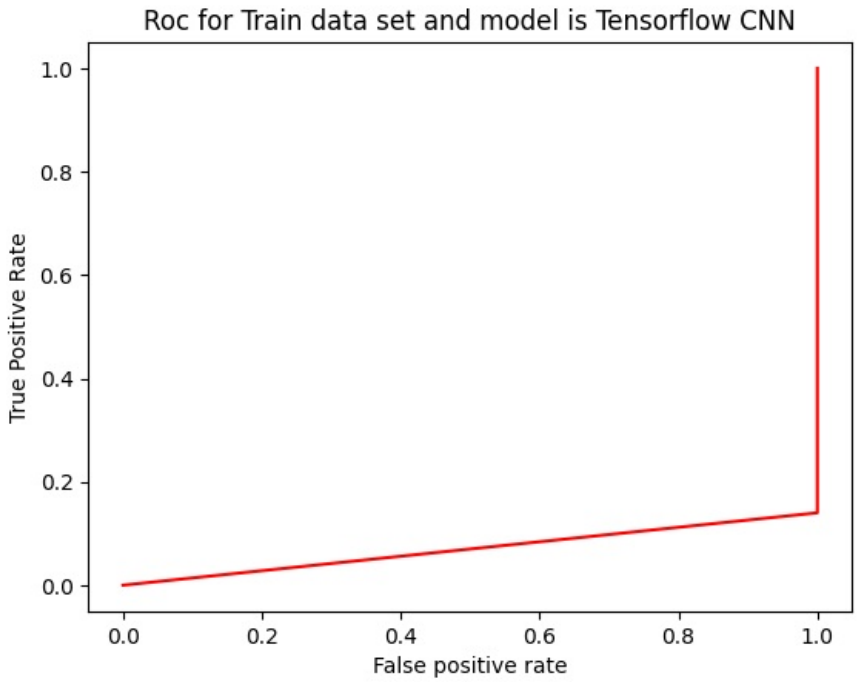
In [59]:
```
test_recall=(cm1[0][0])/((cm1[0][0])+(cm1[0][1]))
test_x=round(test_recall,2)
test_fpr=(cm1[1][0])/((cm1[1][0])+(cm1[1][1]))
test_y=round(test_fpr,2)
```

In [60]:
```
plt.title("Roc for Testing  data set and model is Tensorflow CNN")
plt.xlabel("False positive rate")
plt.ylabel("True Positive Rate")
plt.plot([0,test_x,1],[0,test_y,1],color="green")
plt.show()
```

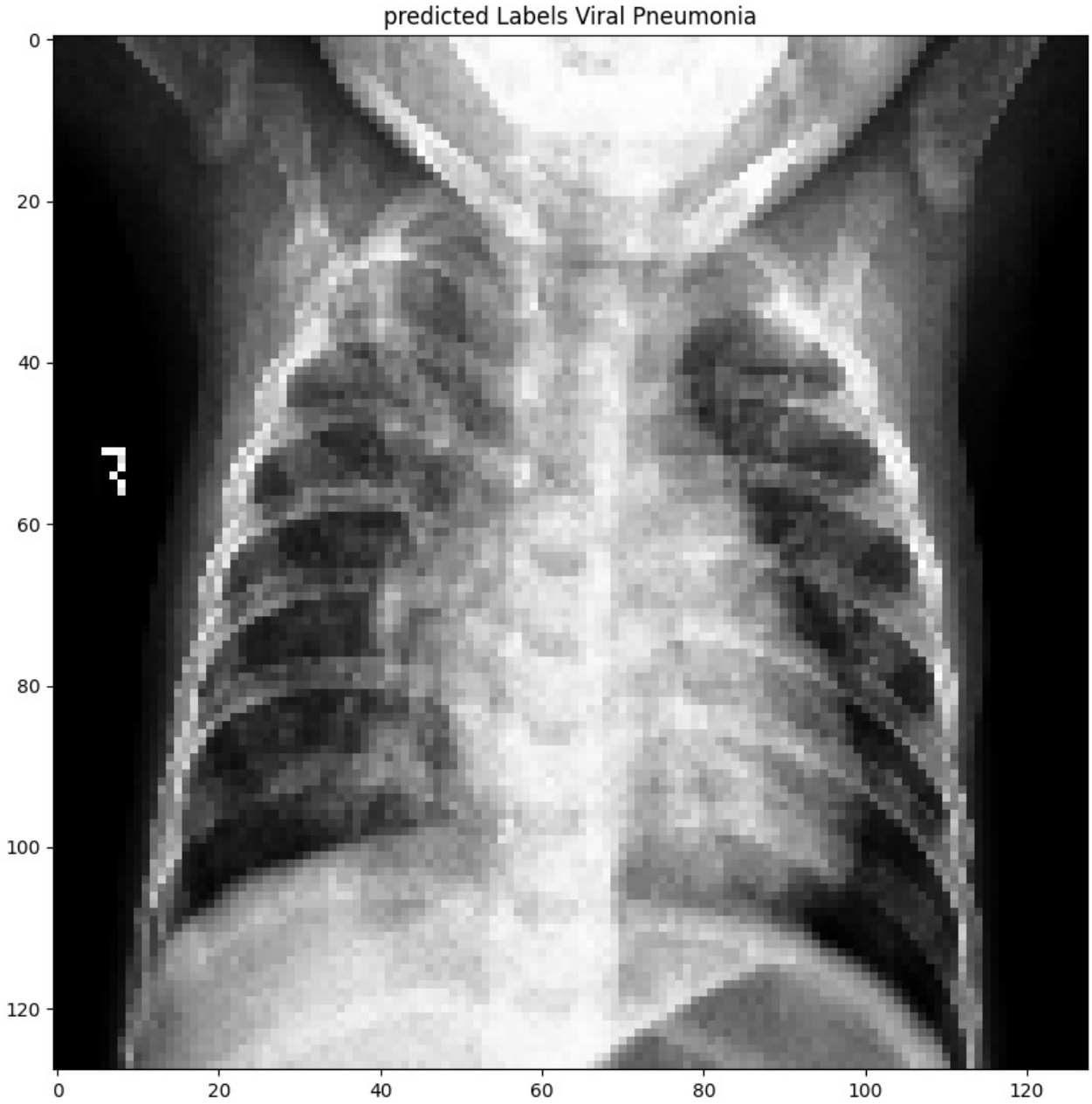


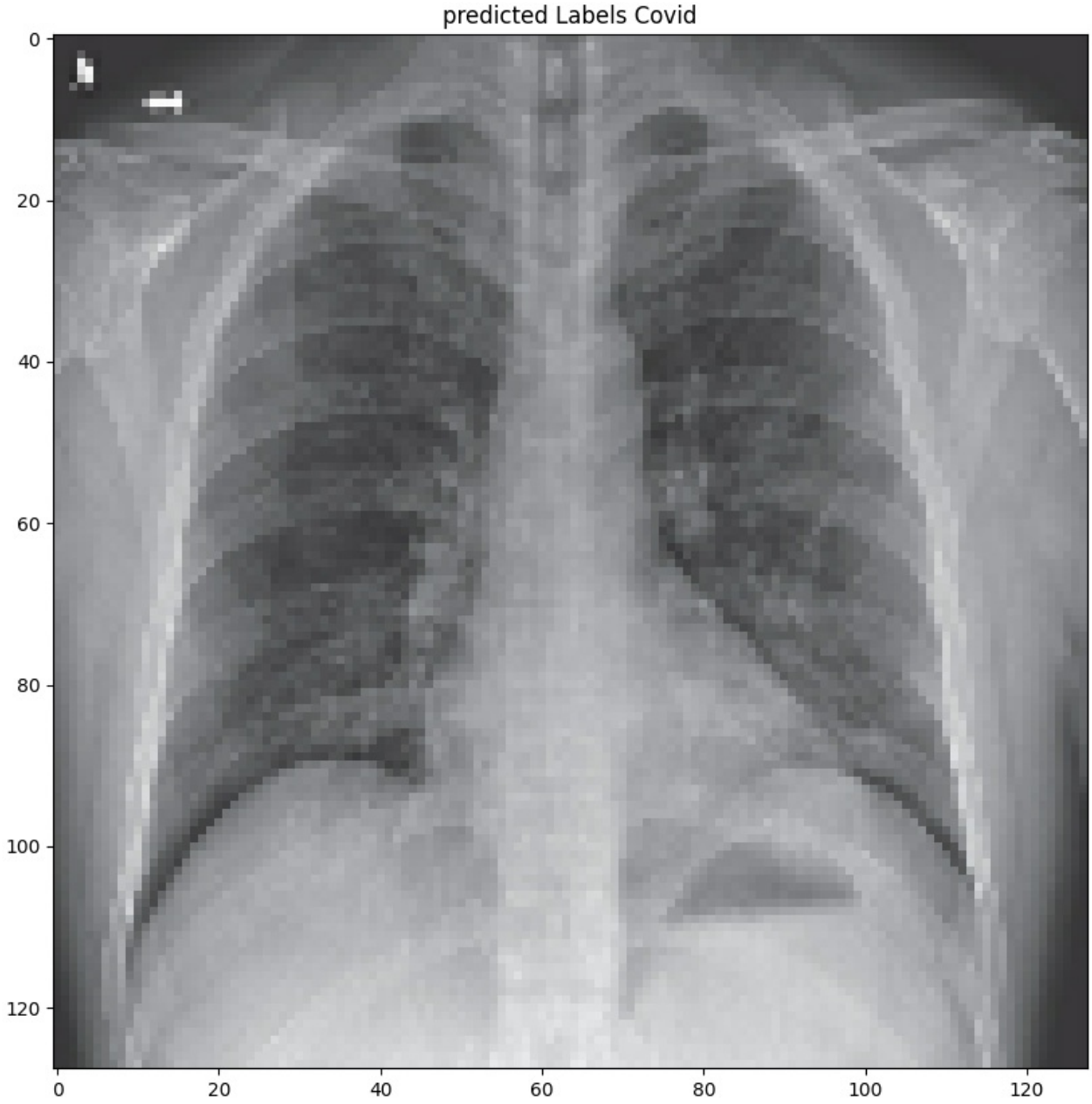
In [61]:
```
train_recall=(cm_train[0][0])/((cm_train[0][0])+(cm_train[0][1]))
train_x=round(train_recall,2)
train_fpr=(cm_train[1][0])/((cm_train[1][0])+(cm_train[1][1]))
train_y=round(train_fpr,2)
```

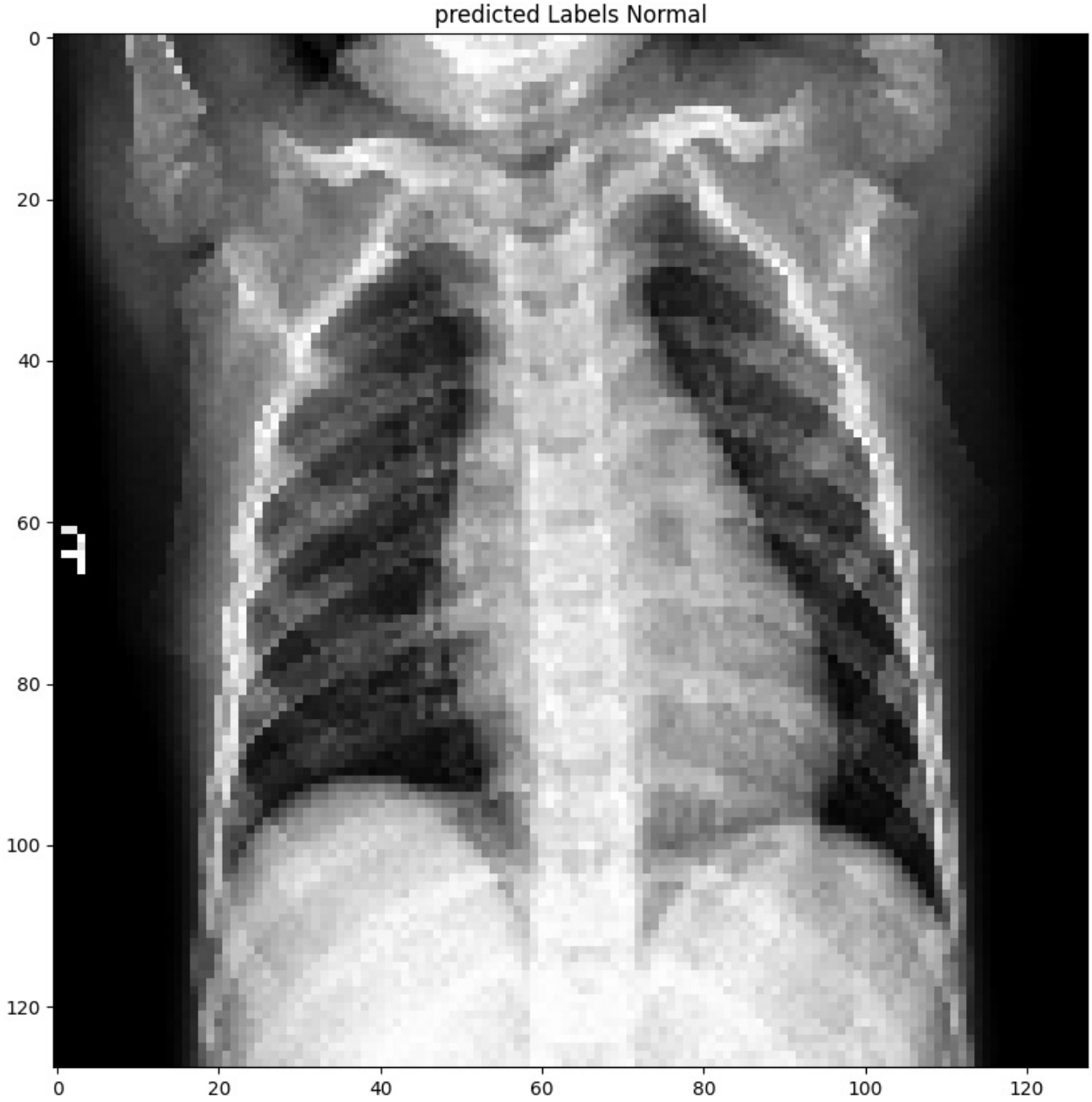
In [62]:
```
plt.title("Roc for Train data set and model is Tensorflow CNN")
plt.xlabel("False positive rate")
plt.ylabel("True Positive Rate")
plt.plot([0,train_x,1],[0,train_y,1],color="red")
plt.show()
```

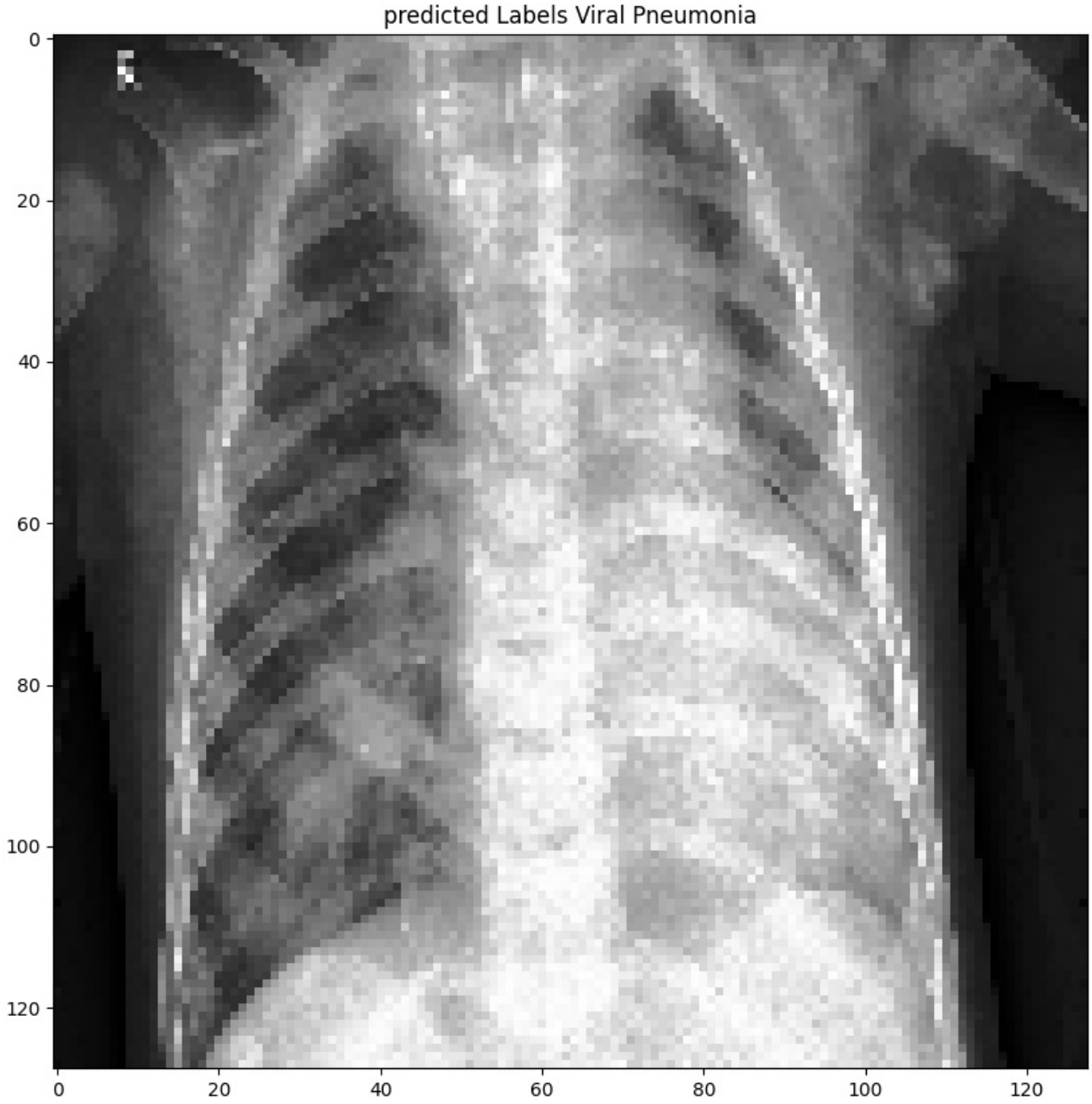


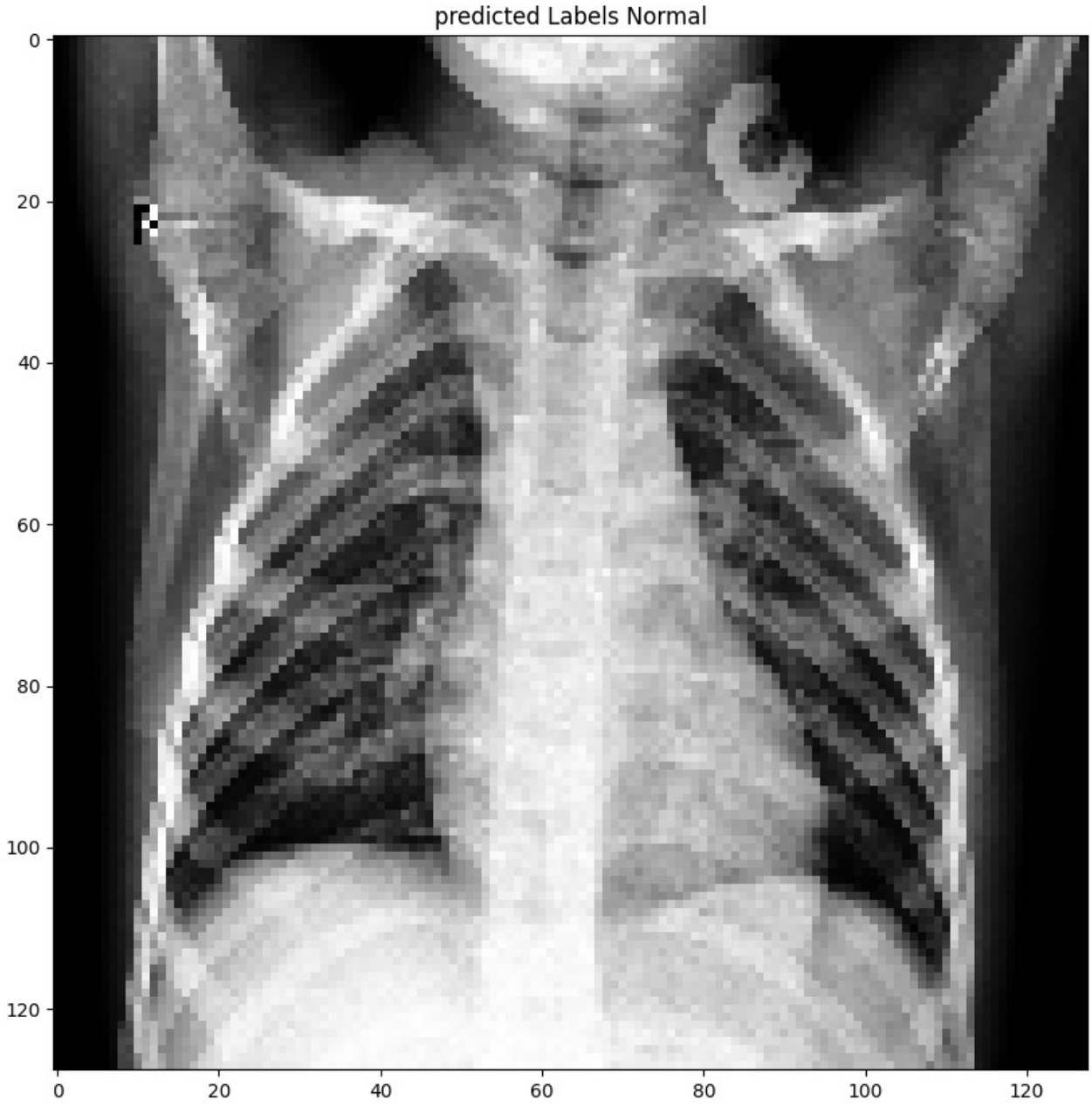
In [71]:
```
#VISUALIZING TEST IMAGES AND PREDICTED CLASSES
for i in range(len(test_classes)):
    plt.figure(figsize=(10,10))
    plt.title("predicted Labels {}".format(labeling[predicted[i]]))
    plt.imshow(test_images[i])
    plt.show()
```

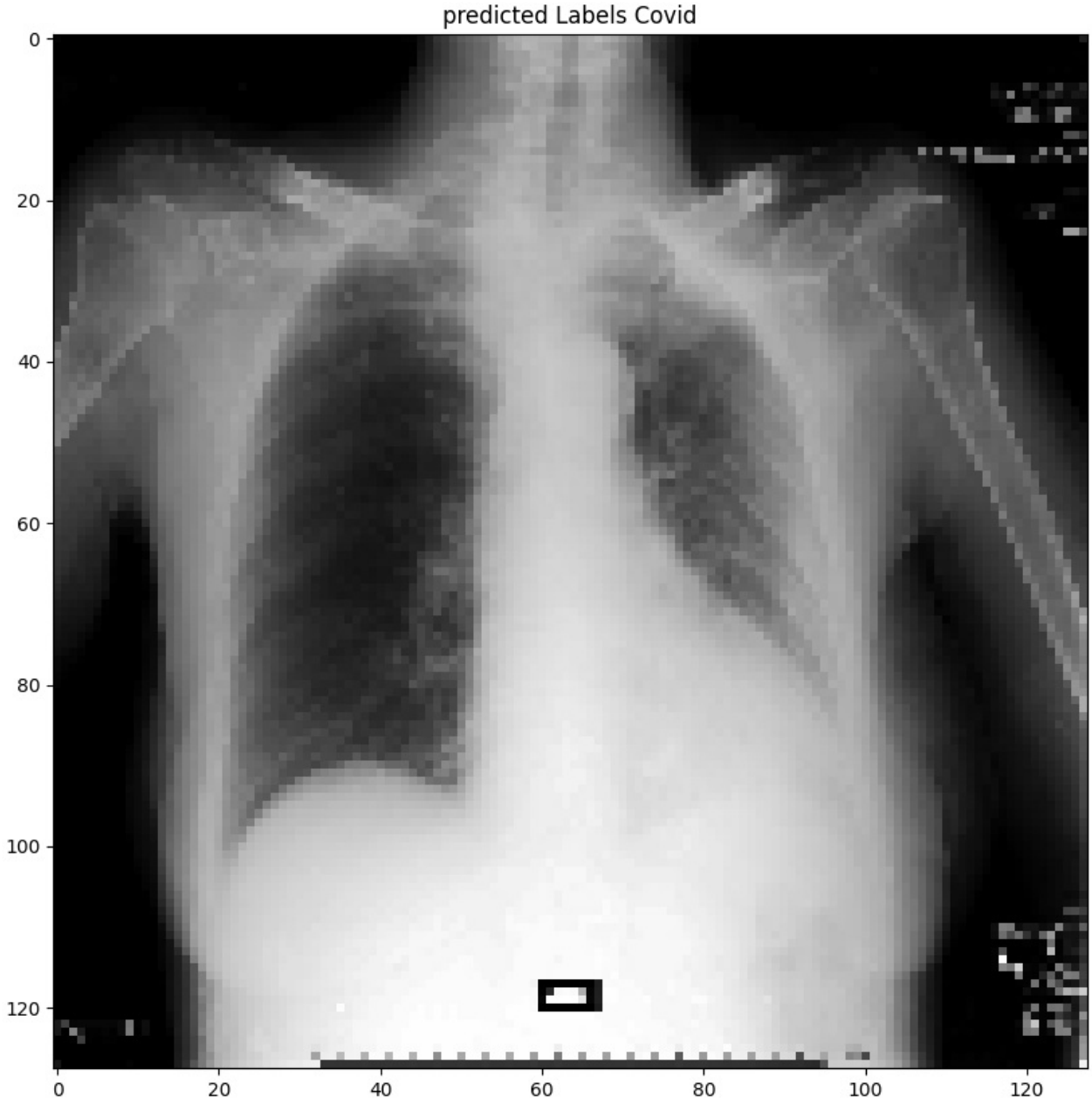

predicted Labels Viral Pneumonia
0
20
40
60
80
100
120
0
20
40
60
80
100
120


predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

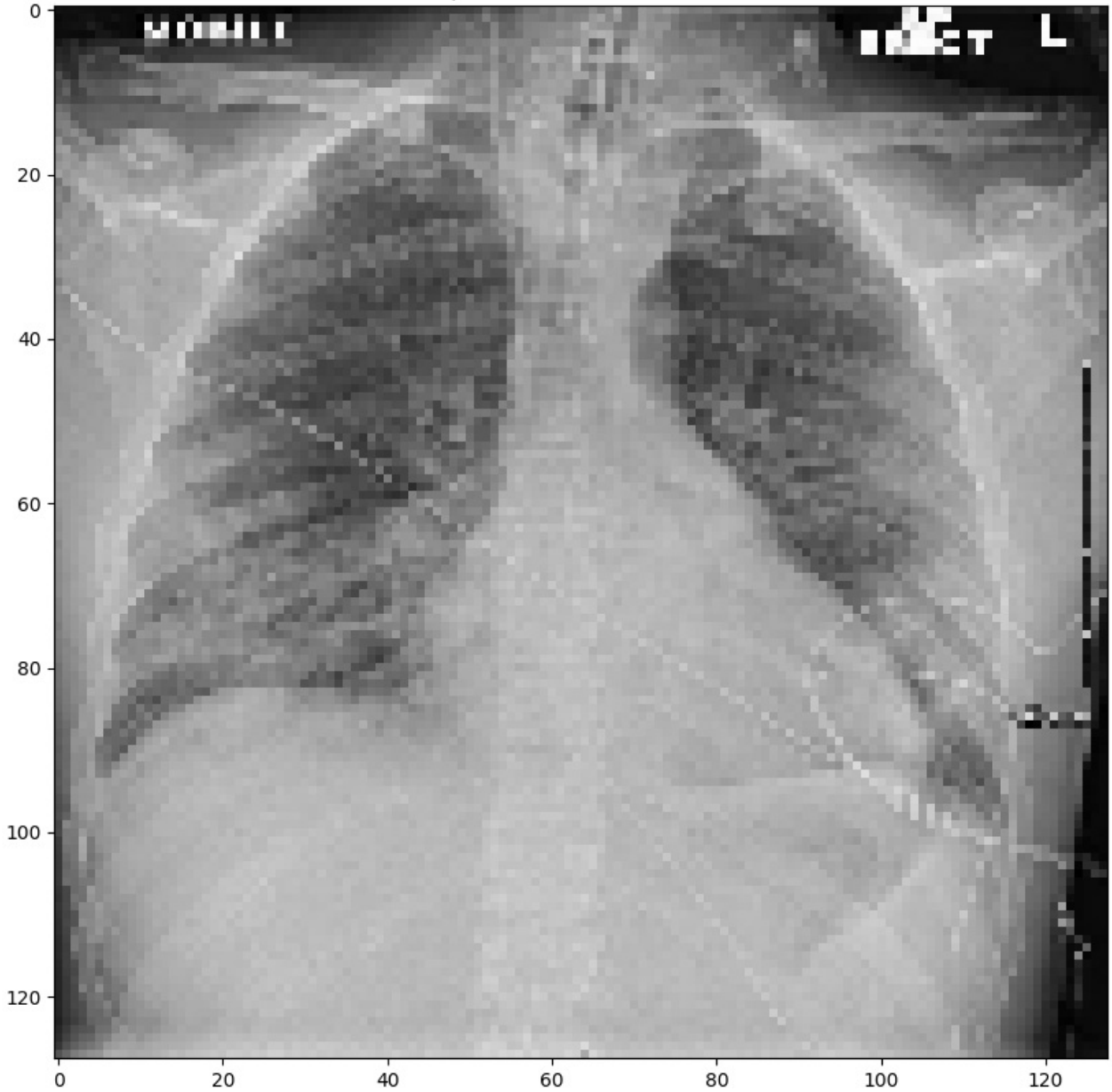

predicted Labels Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

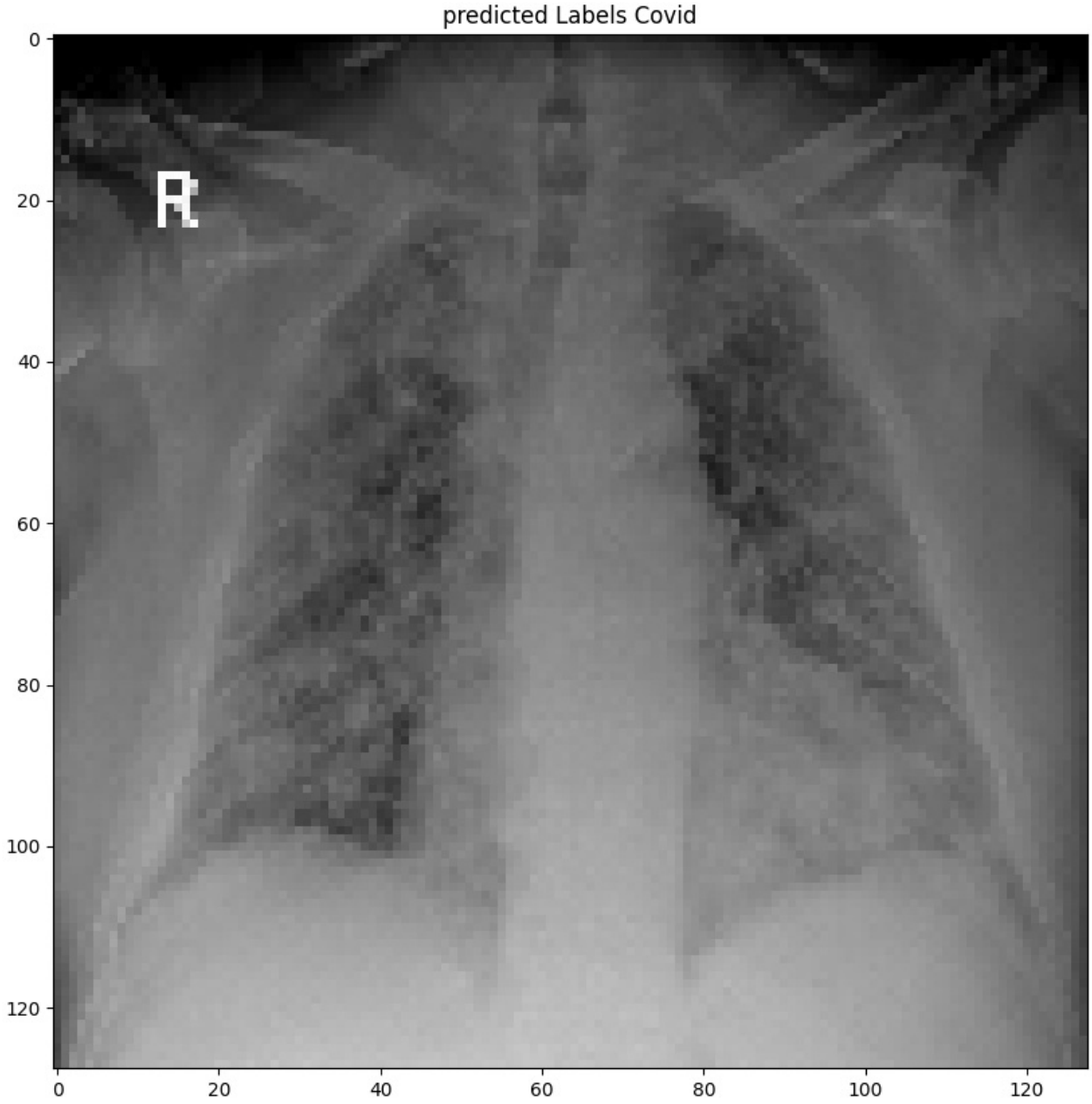

predicted Labels Viral Pneumonia
0
20
40
60
80
100
120
0
20
40
60
80
100
120

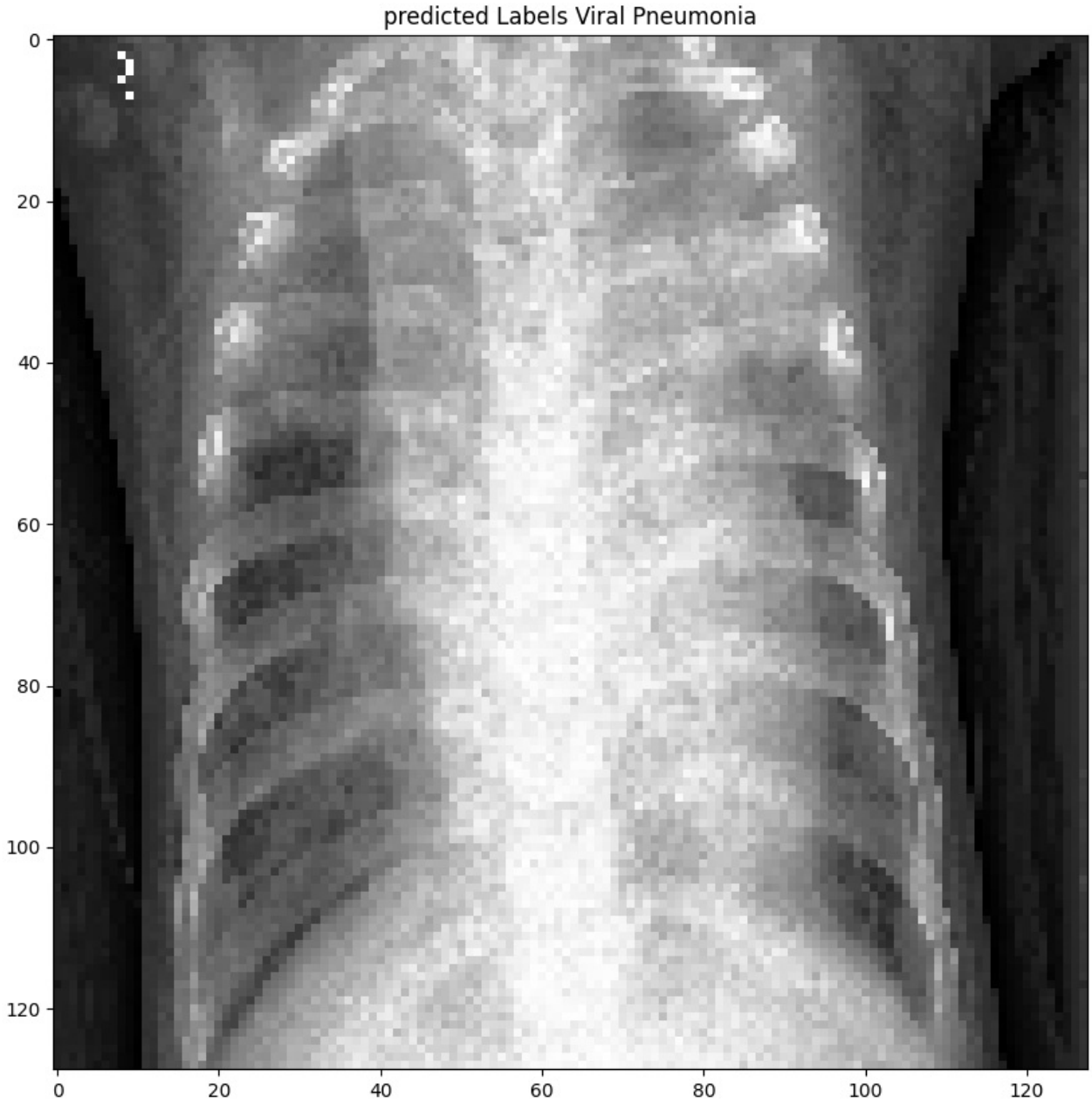

predicted Labels Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

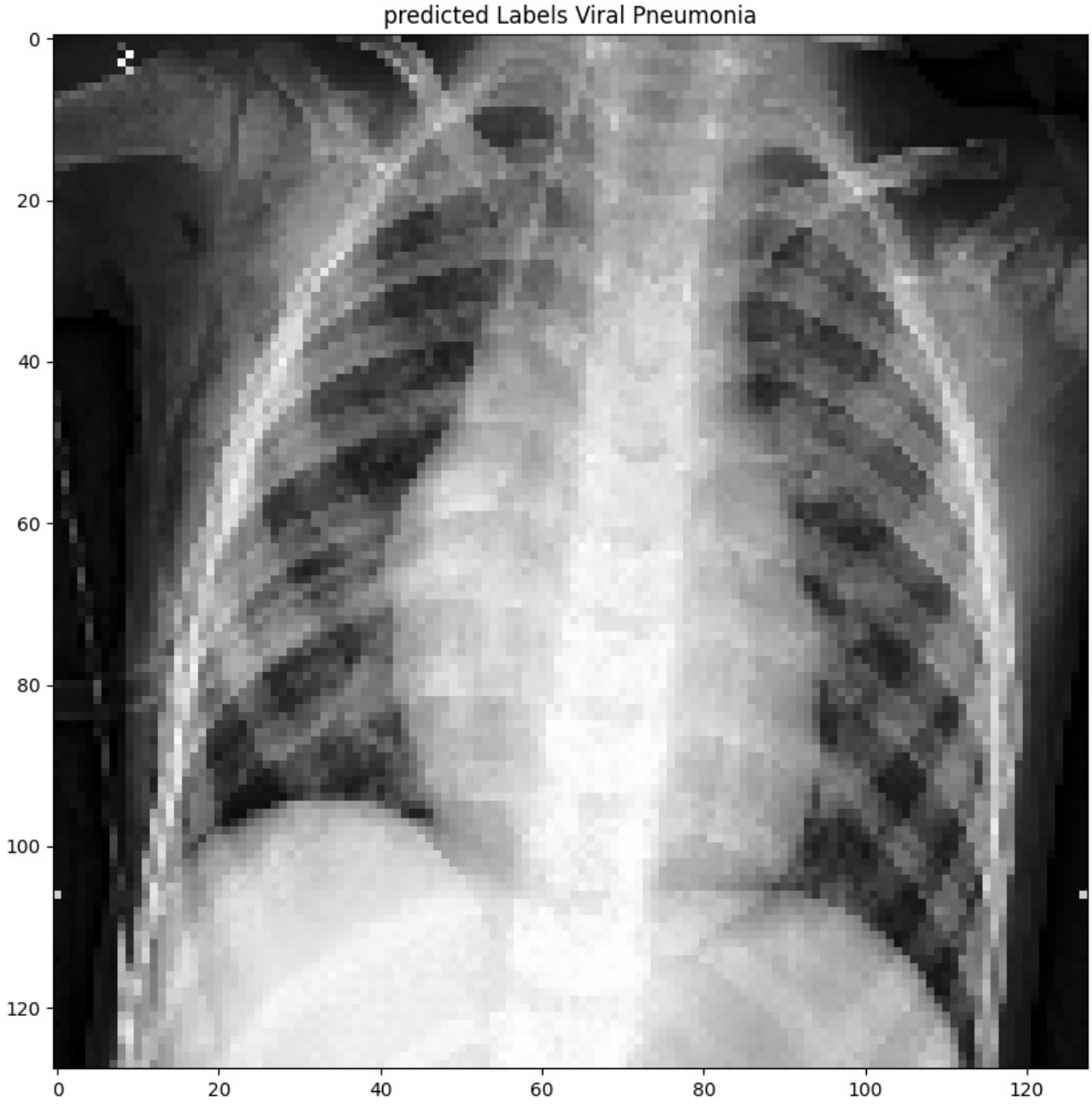

predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

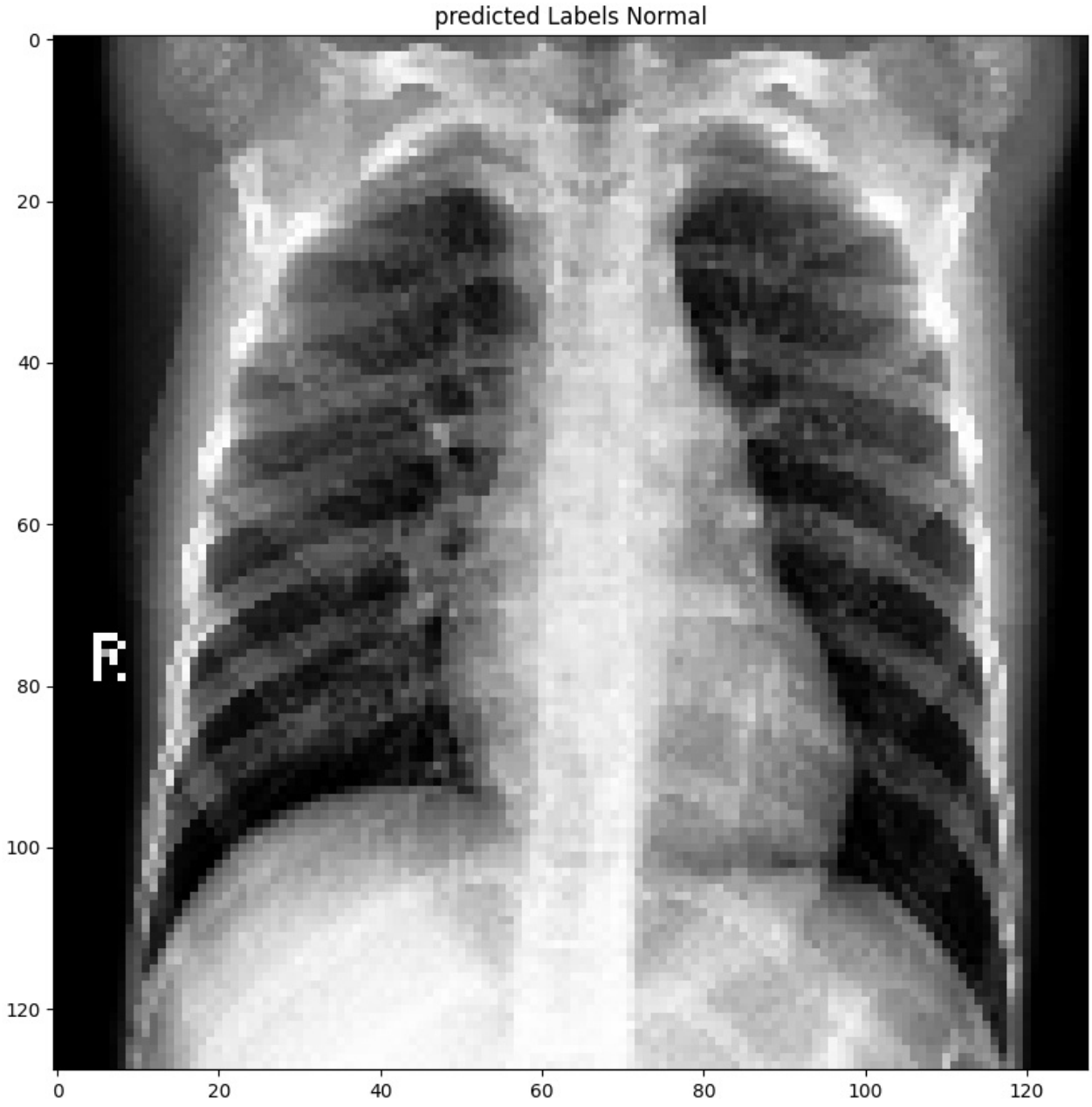
predicted Labels Covid

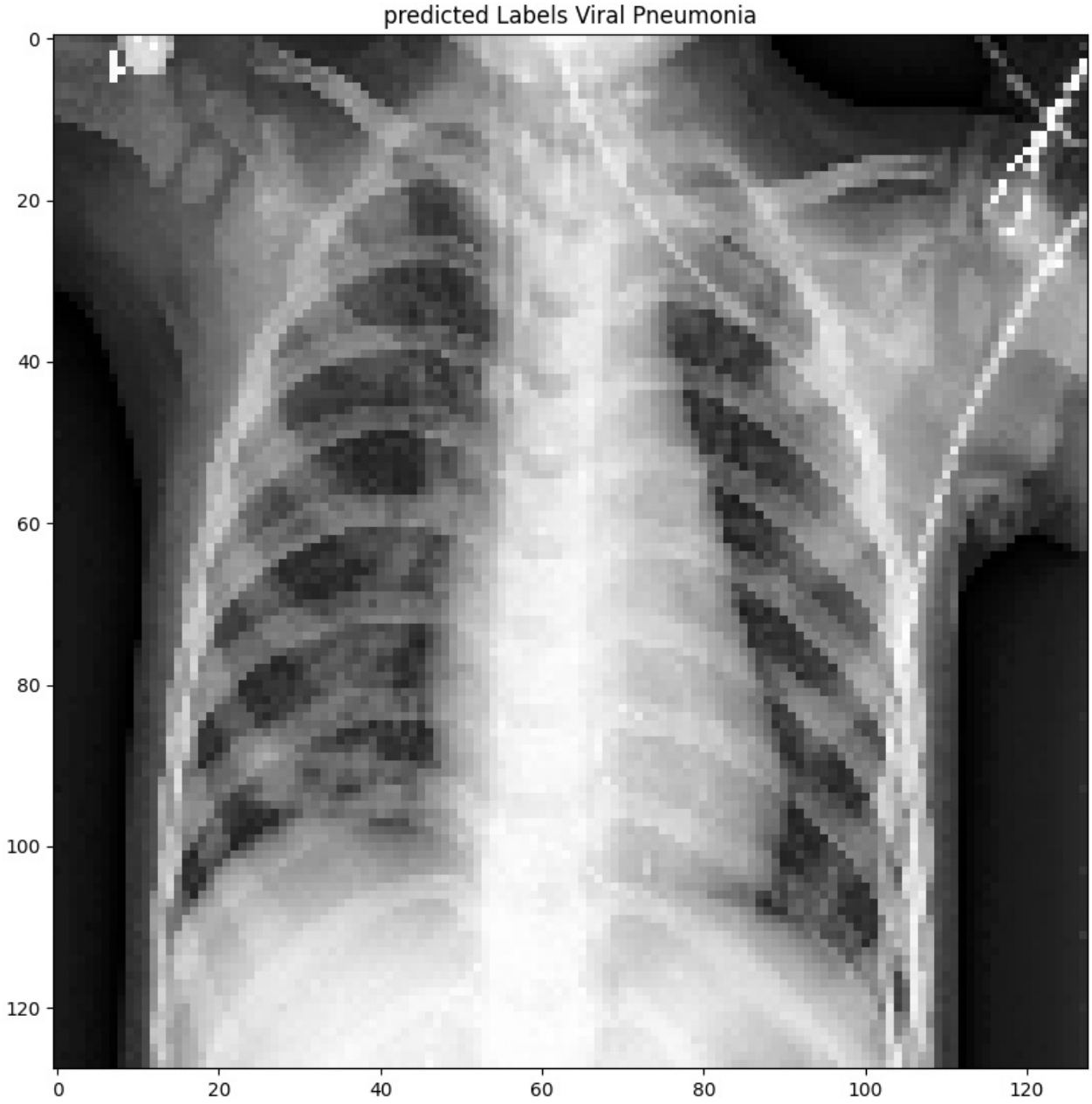


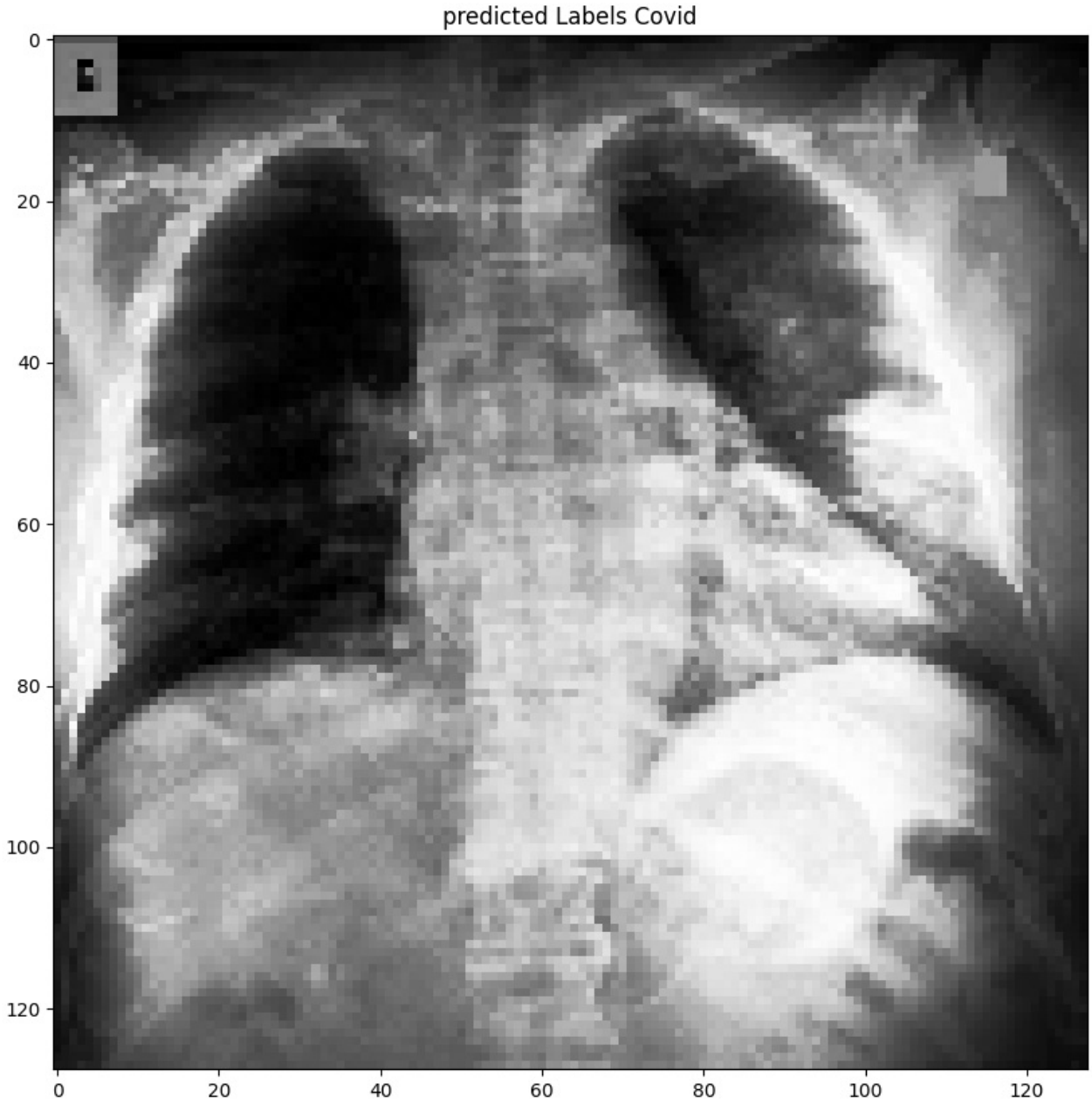

predicted Labels Covid
R
0
20
40
60
80
100
120
0
20
40
60
80
100
120

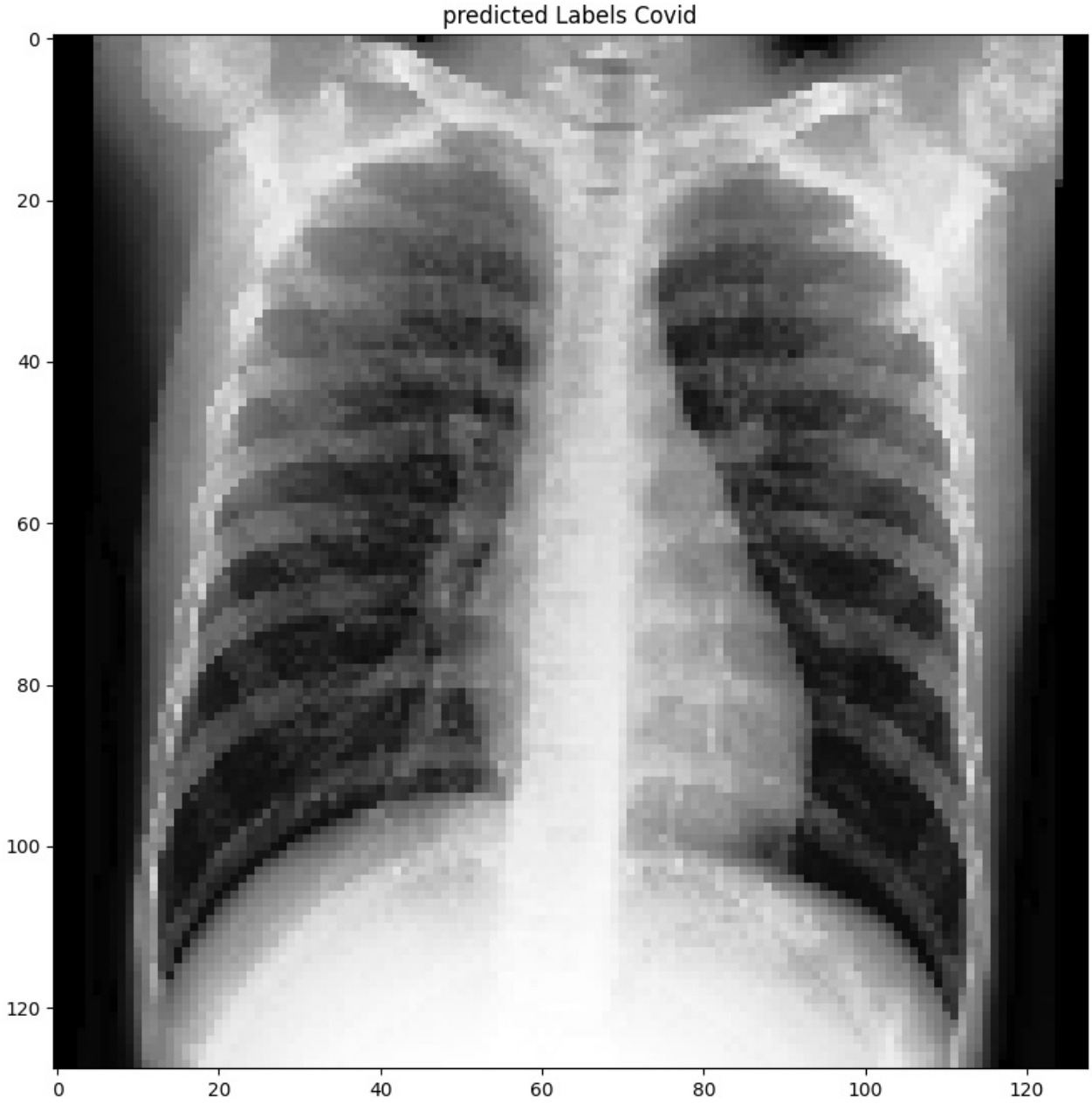

predicted Labels Viral Pneumonia
0
20
40
60
80
100
120
0
20
40
60
80
100
120

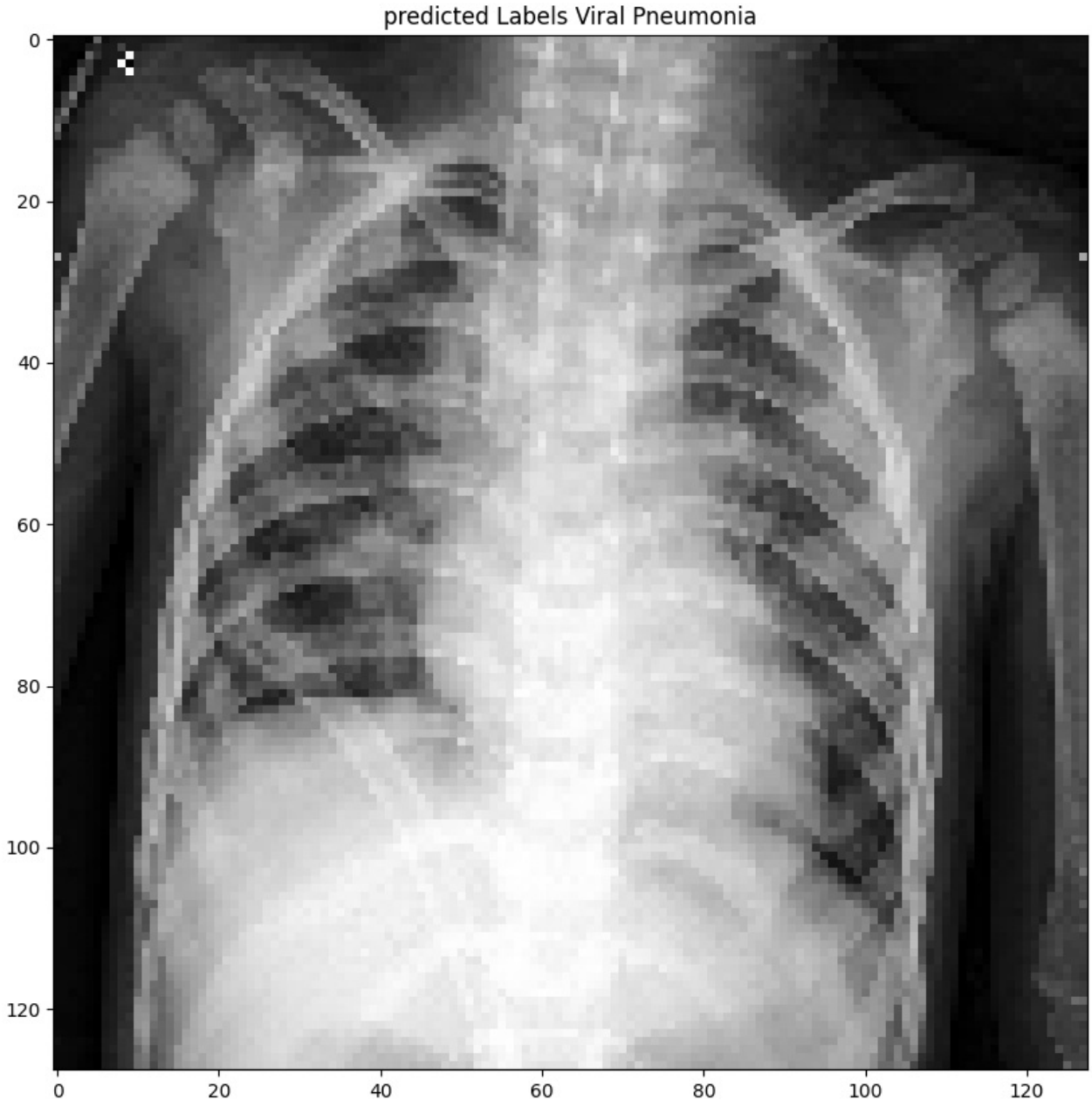

predicted Labels Viral Pneumonia
0
20
40
60
80
100
120
0
20
40
60
80
100
120

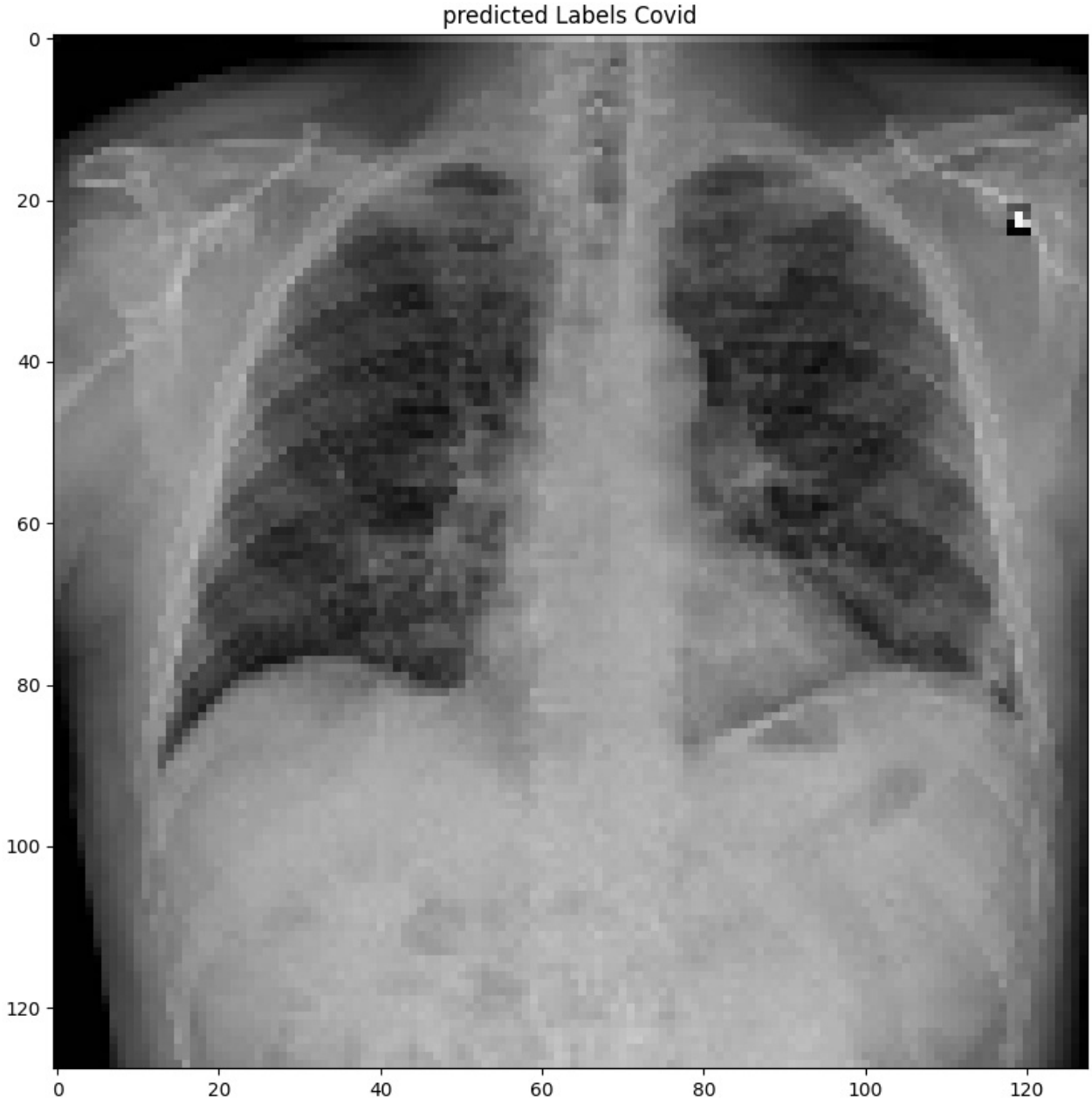

predicted Labels Normal
R
0
20
40
60
80
100
120
0
20
40
60
80
100
120

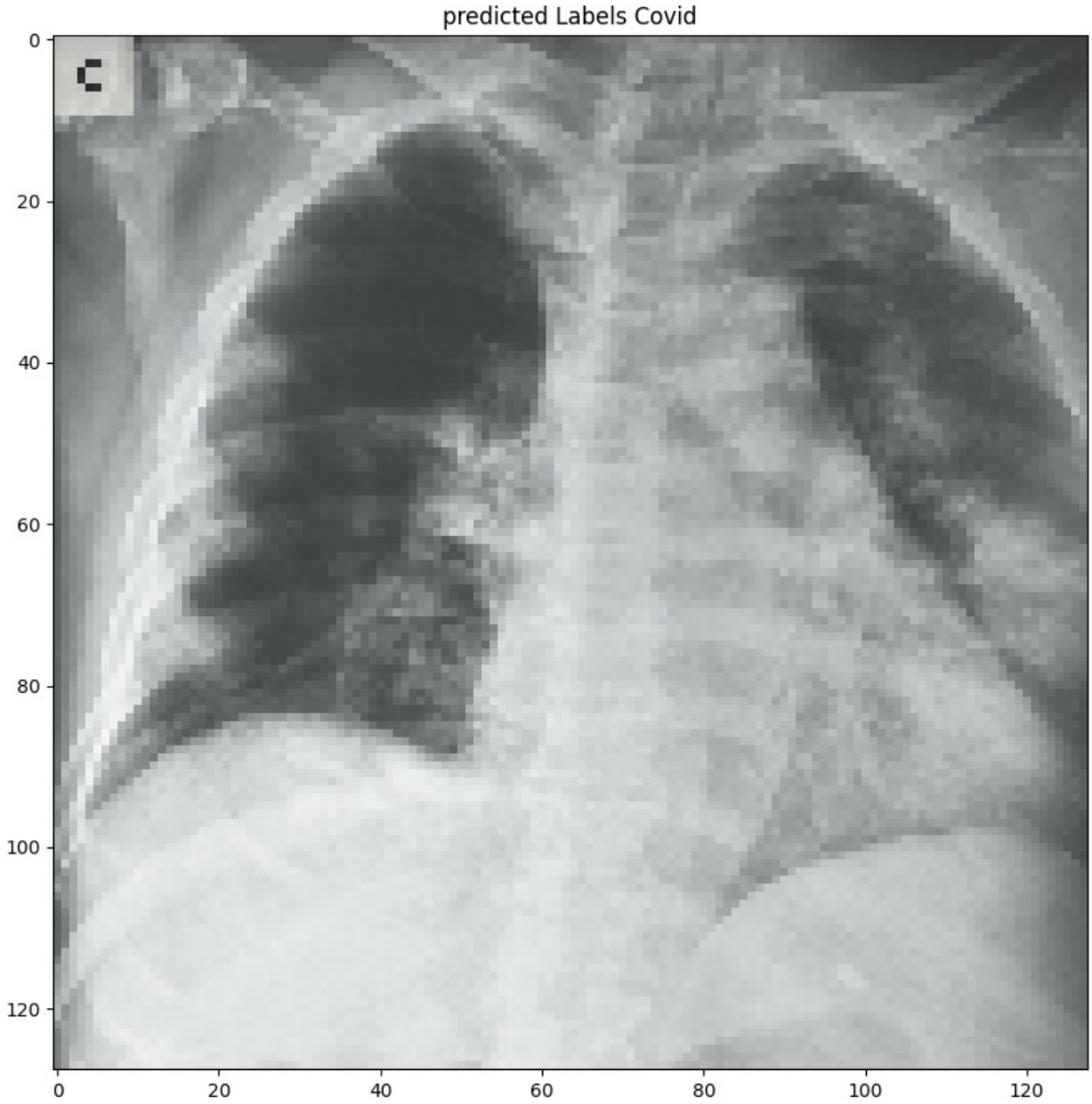

predicted Labels Viral Pneumonia
0
20
40
60
80
100
120
0
20
40
60
80
100
120

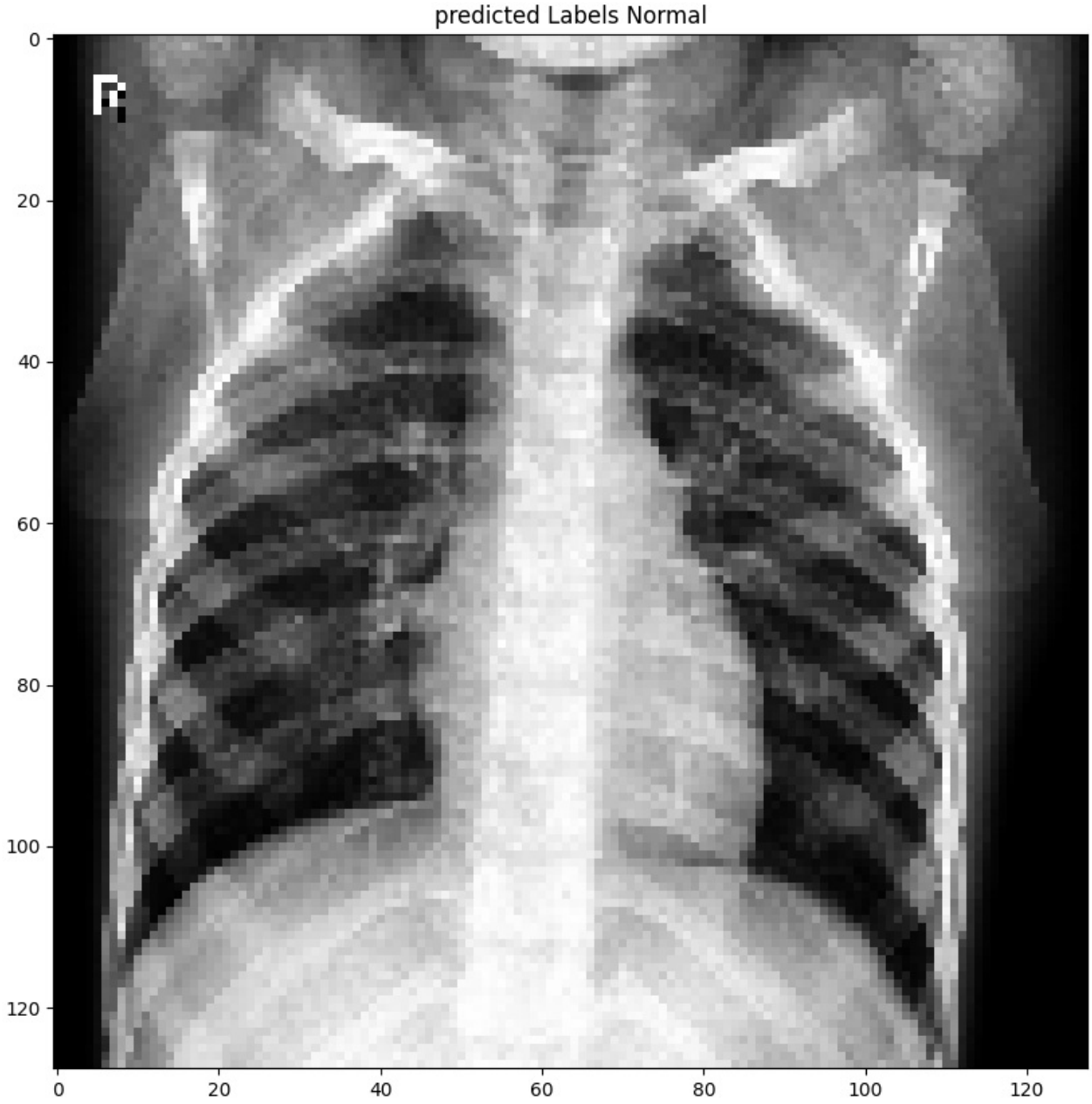

predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

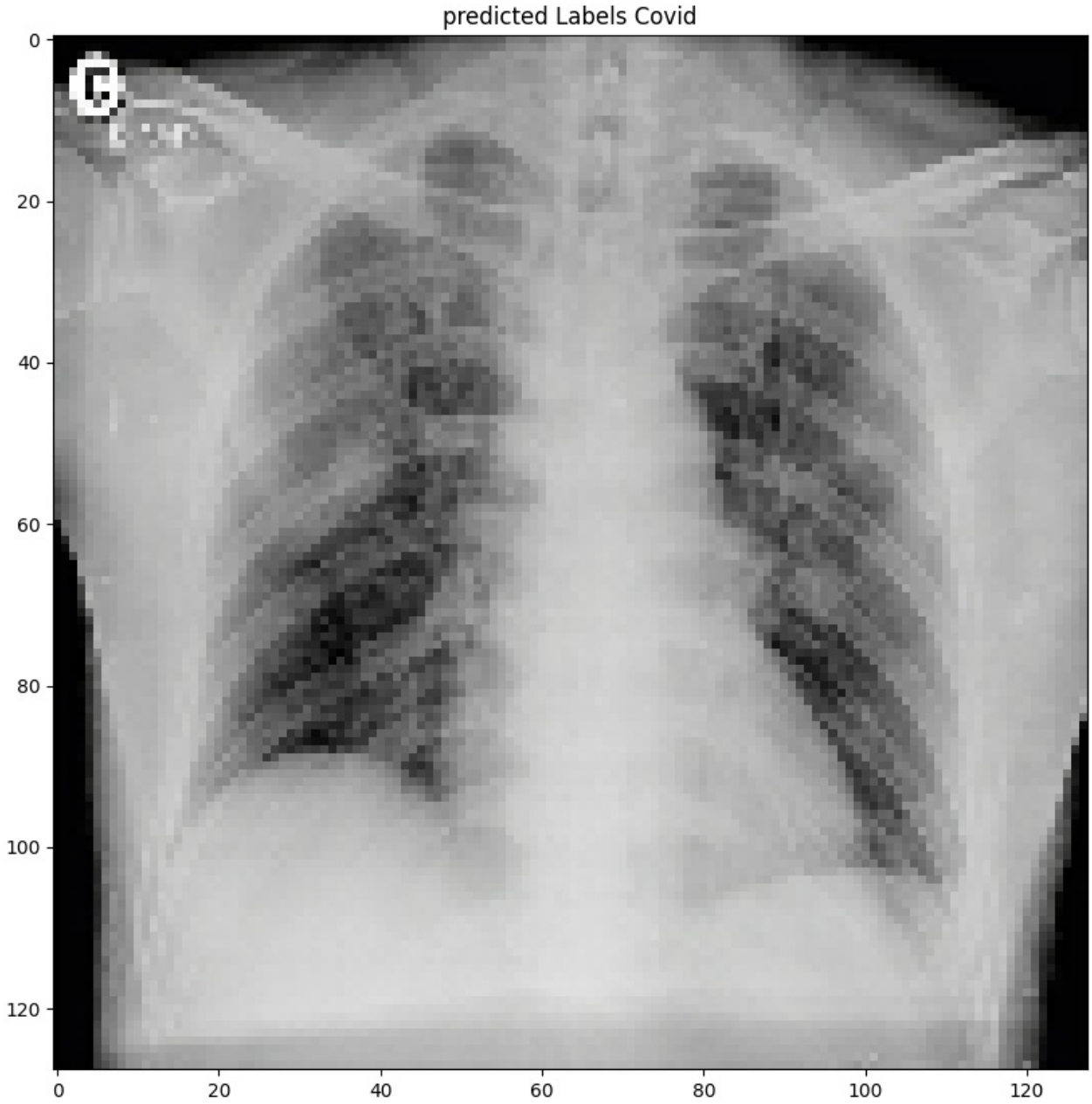

predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

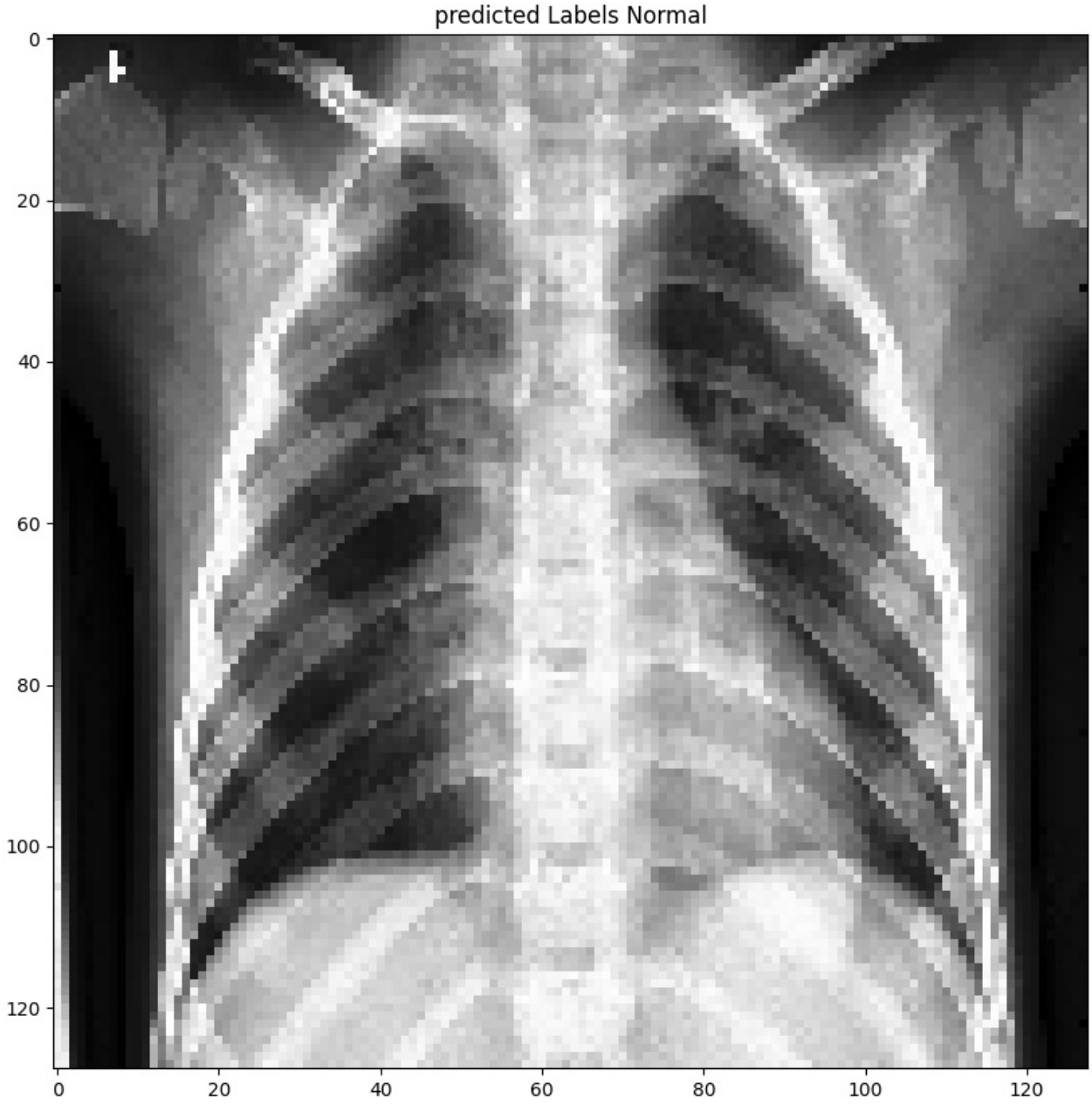

predicted Labels Viral Pneumonia
0
20
40
60
80
100
120
0
20
40
60
80
100
120

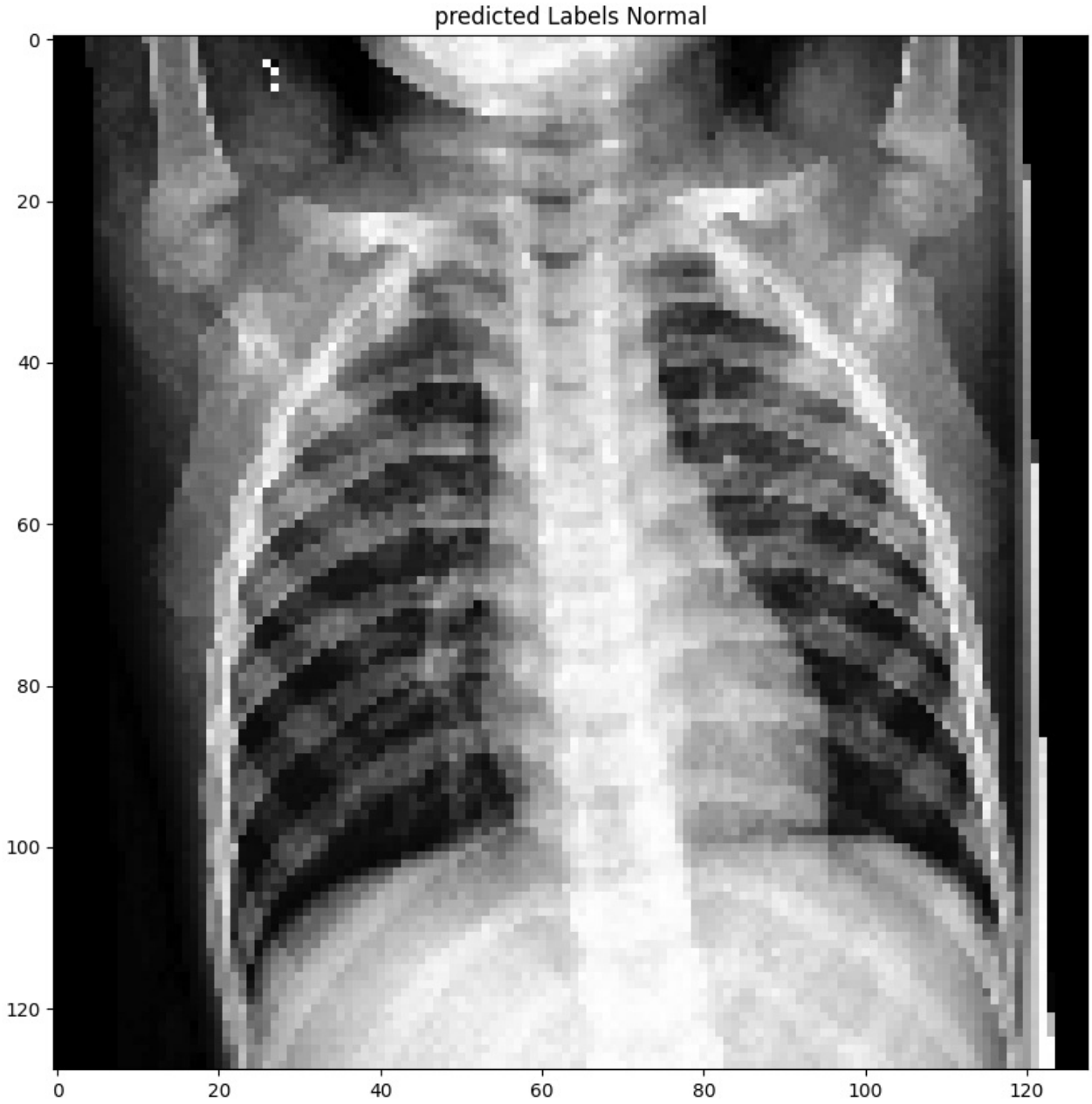

predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

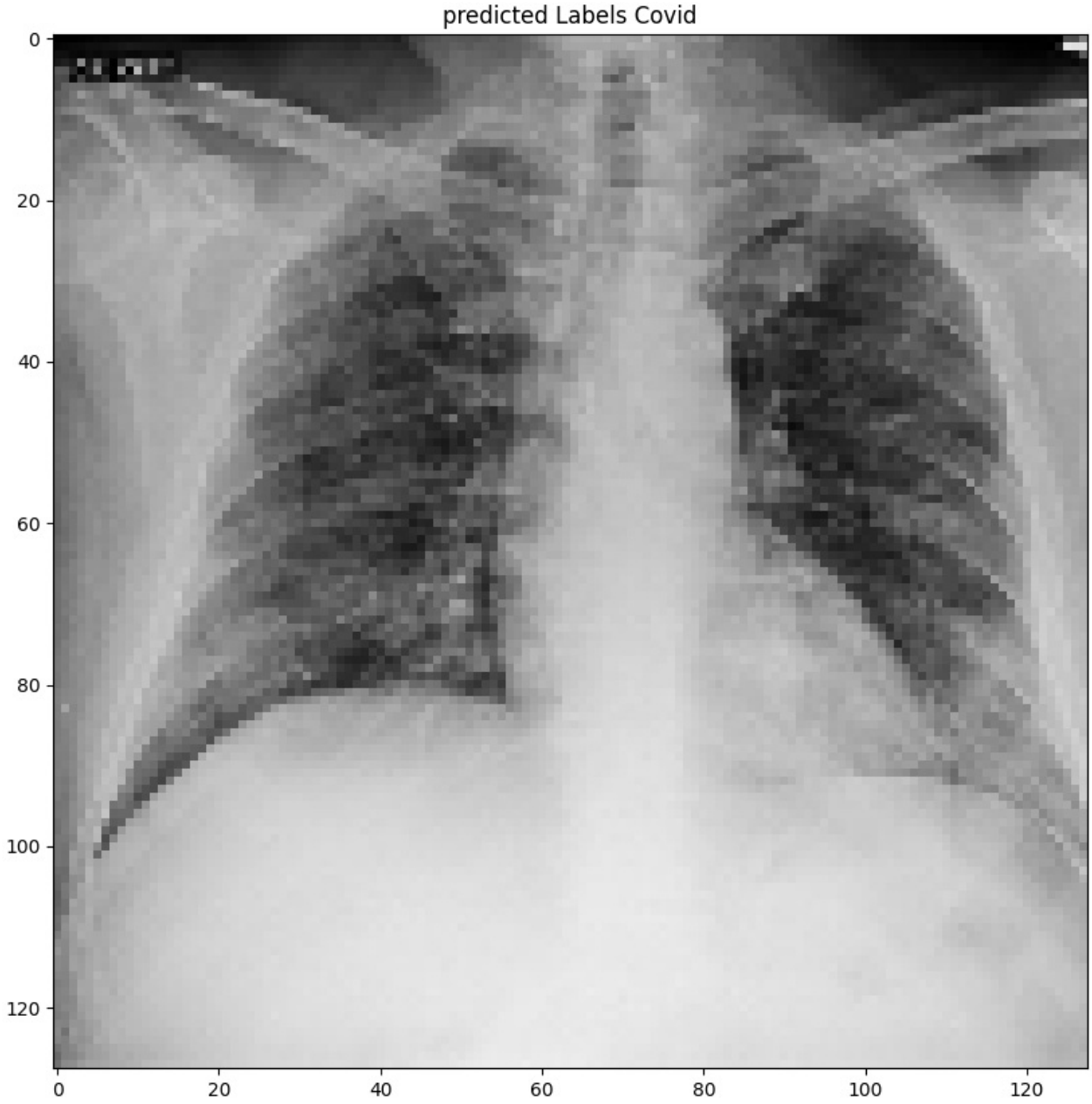

predicted Labels Covid
C
0
20
40
60
80
100
120
0
20
40
60
80
100
120

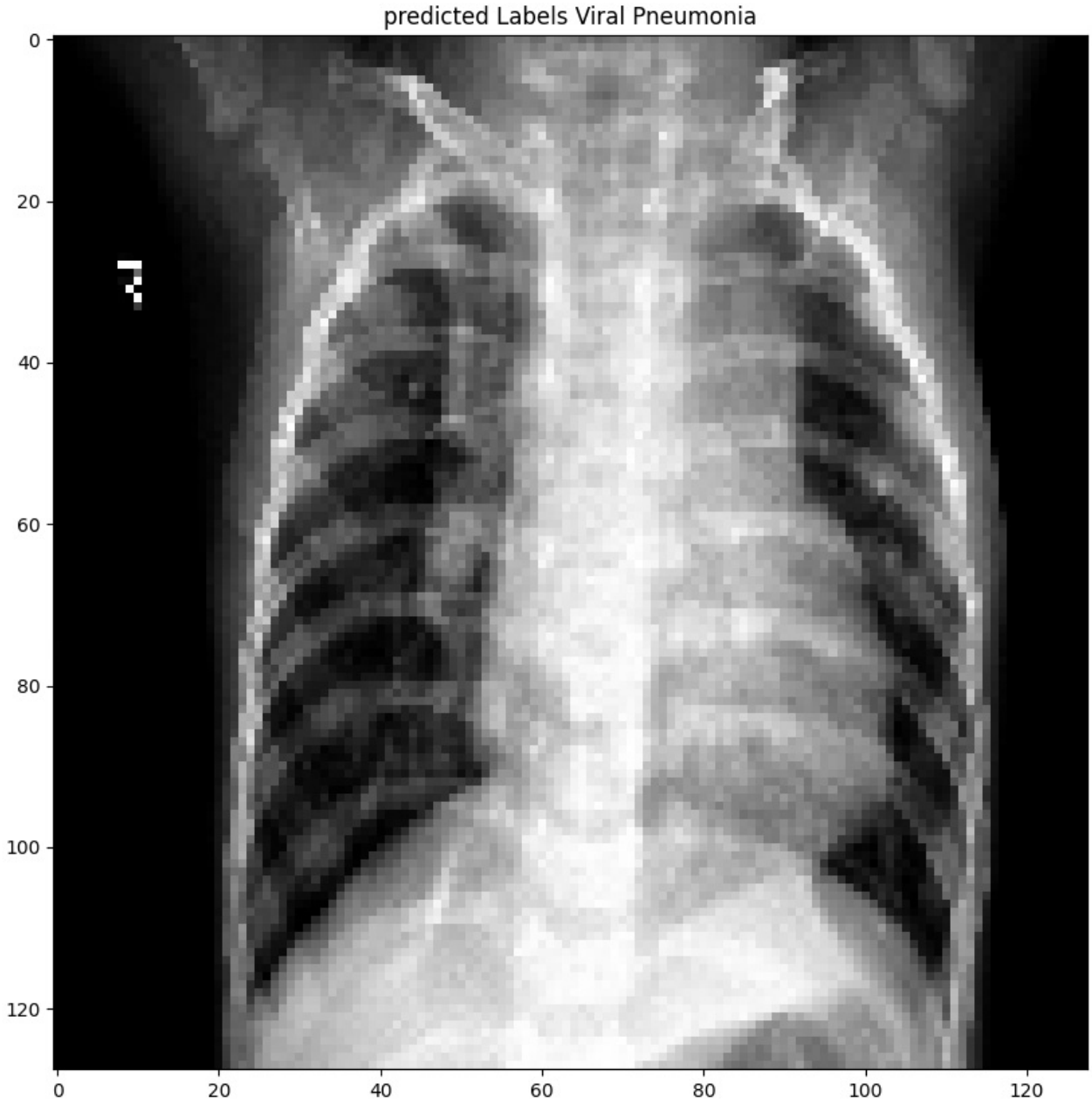

predicted Labels Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

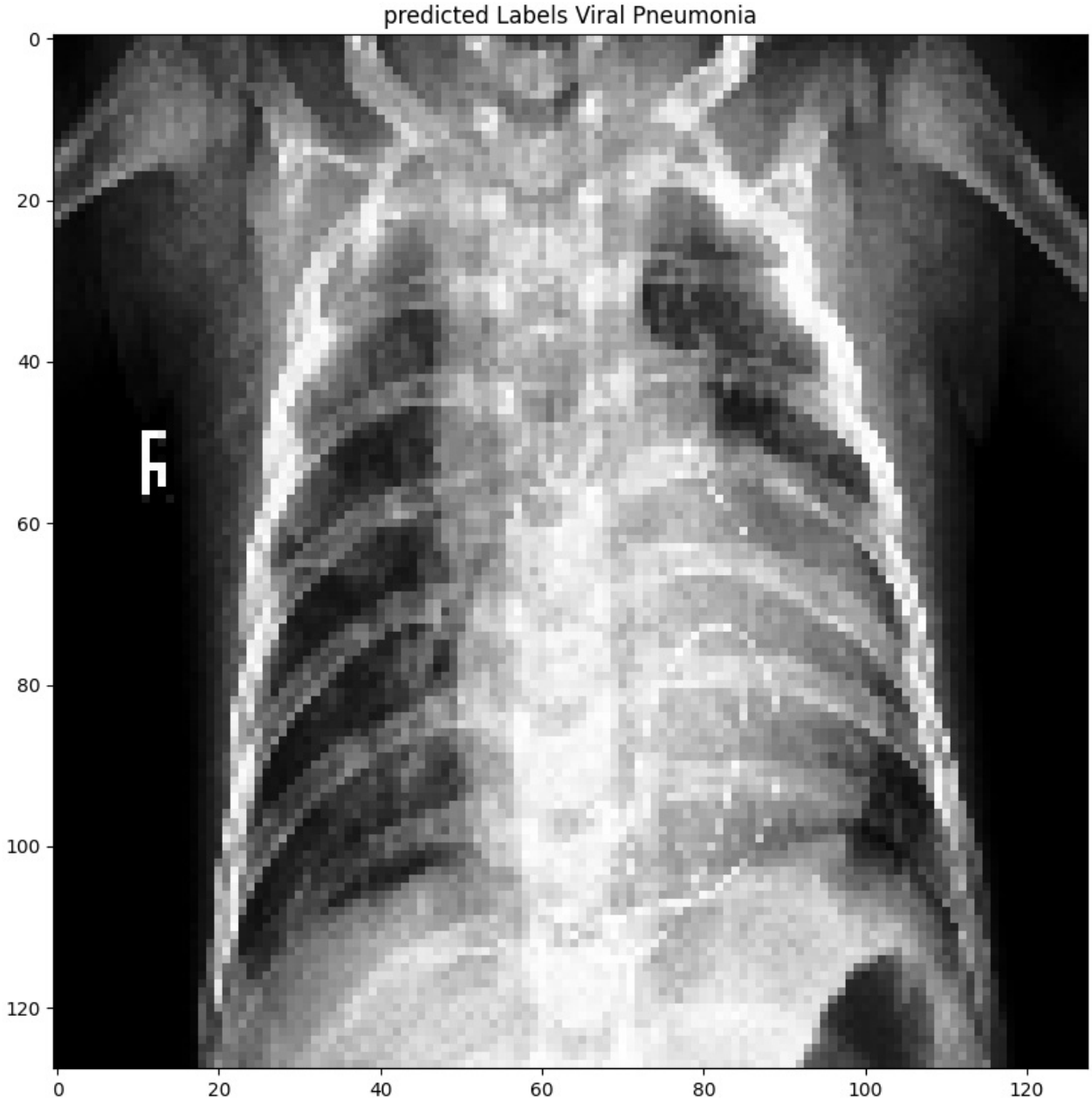

predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

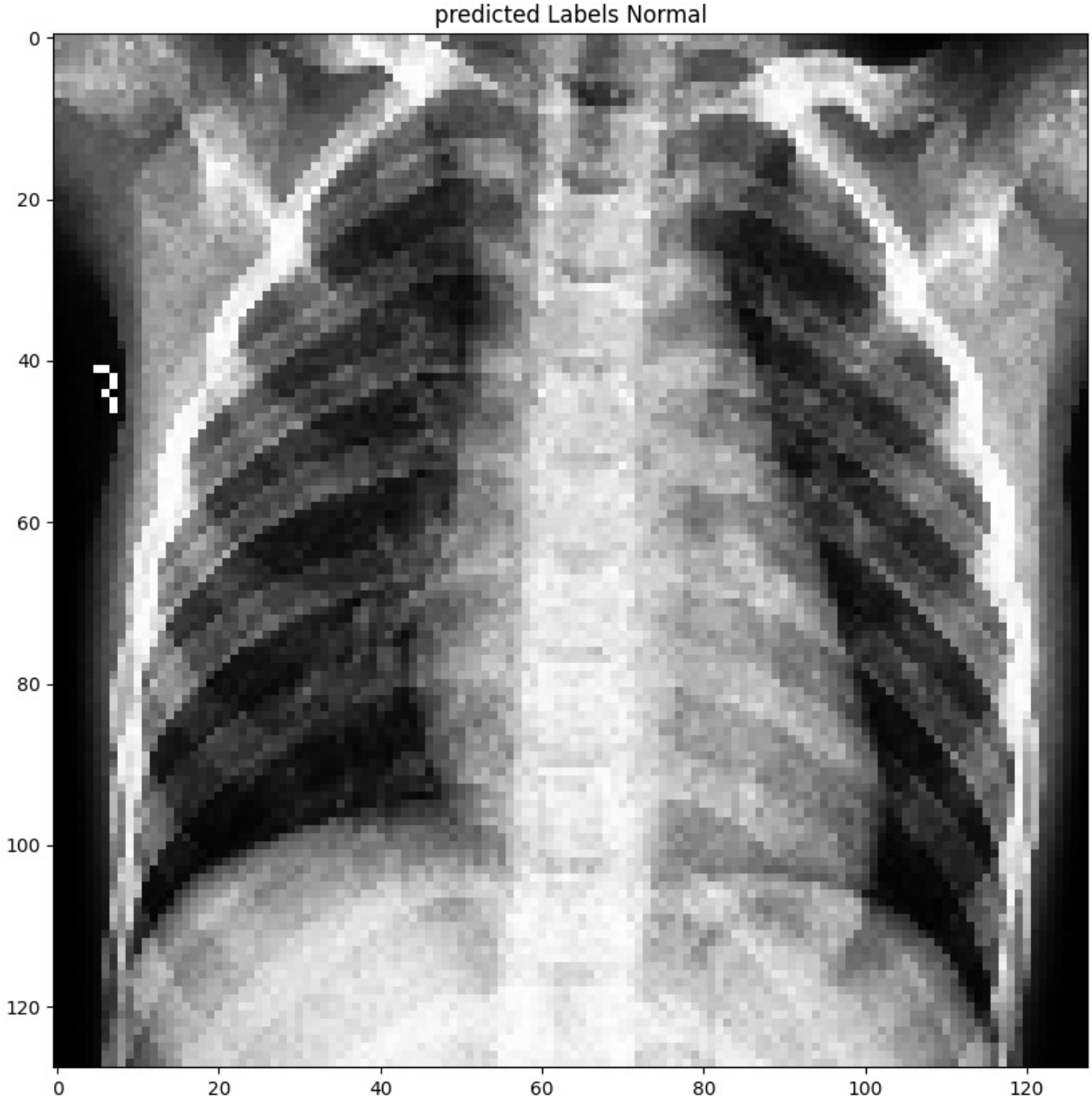

predicted Labels Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

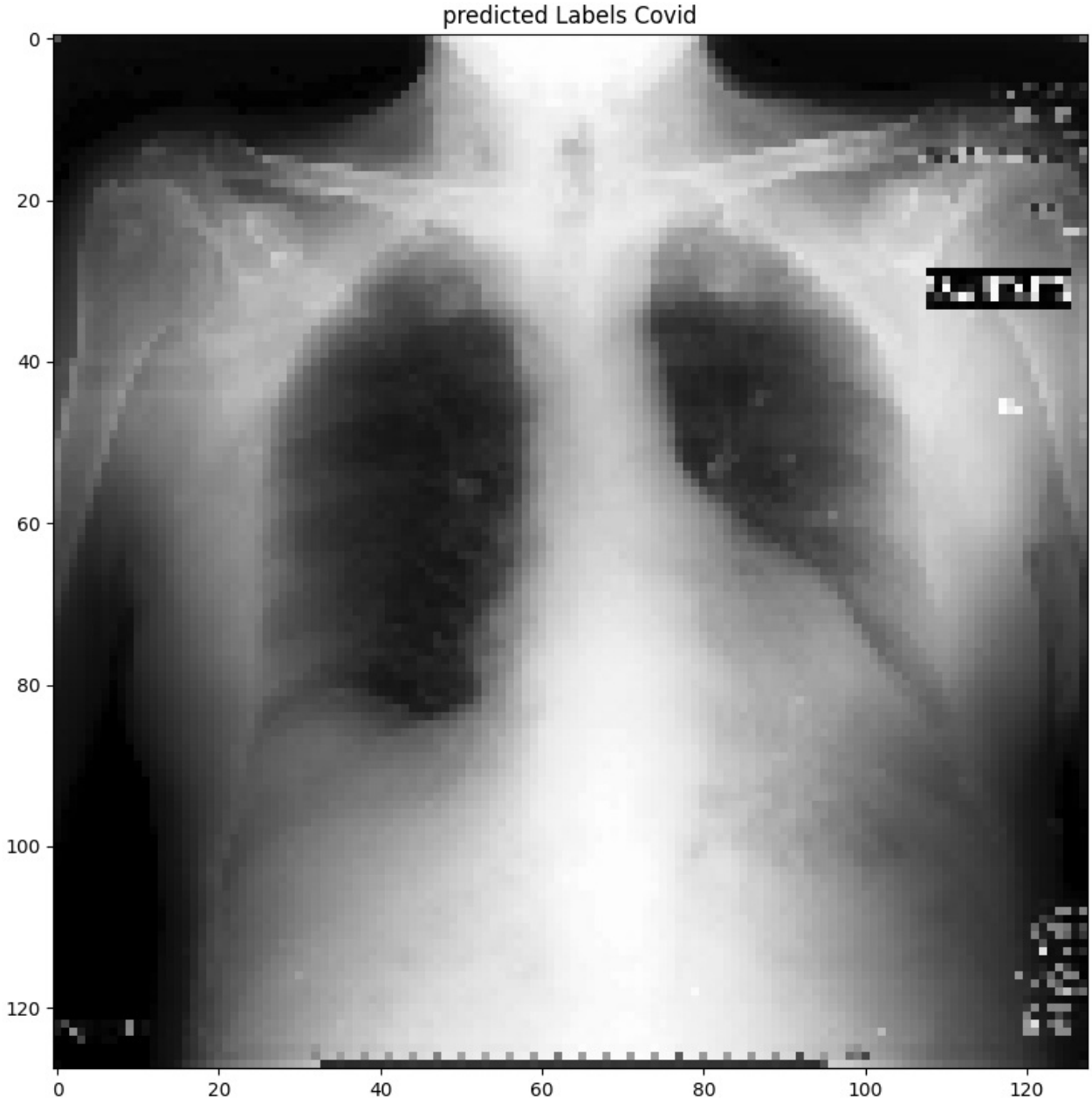

predicted Labels Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

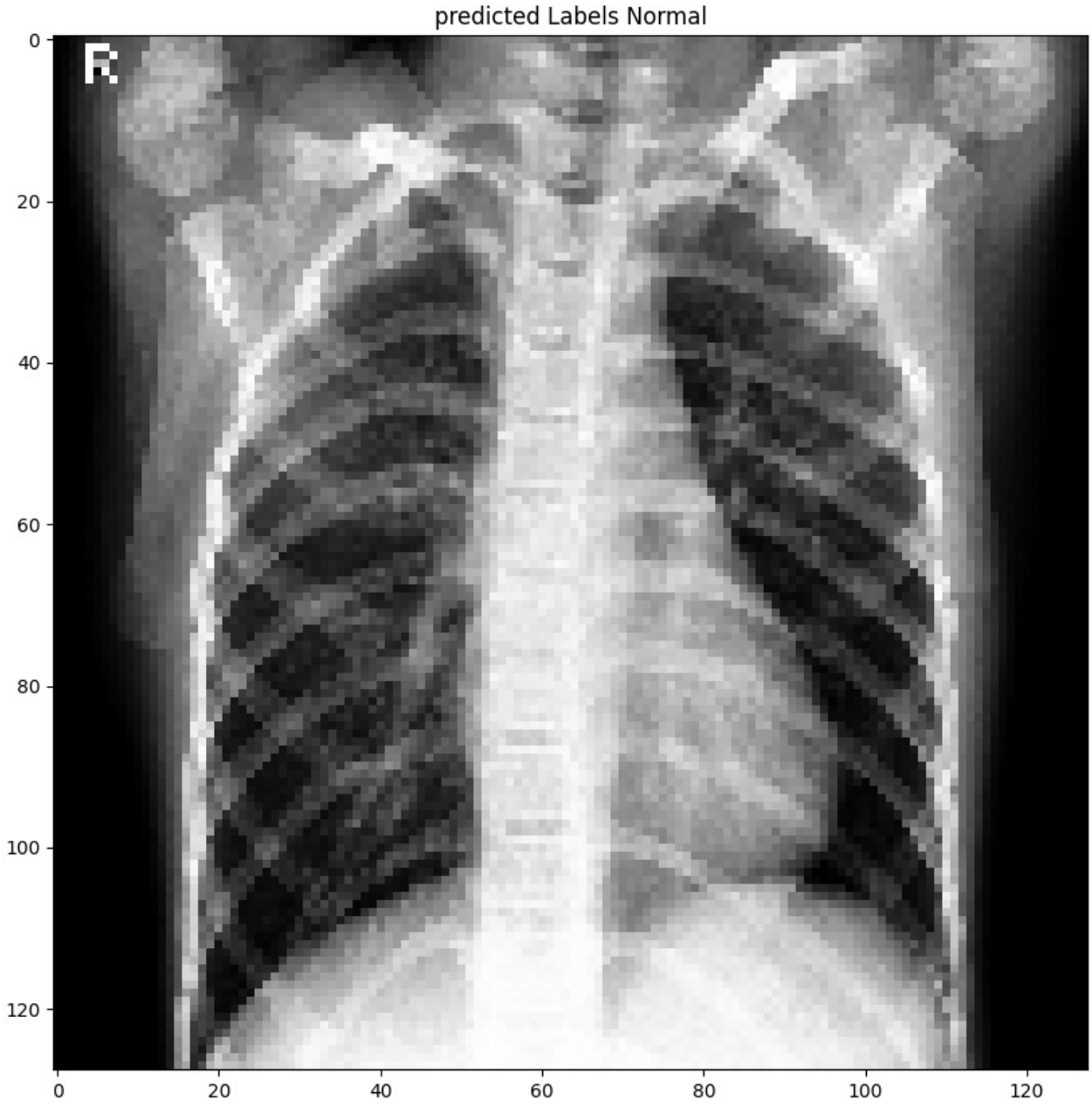

predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

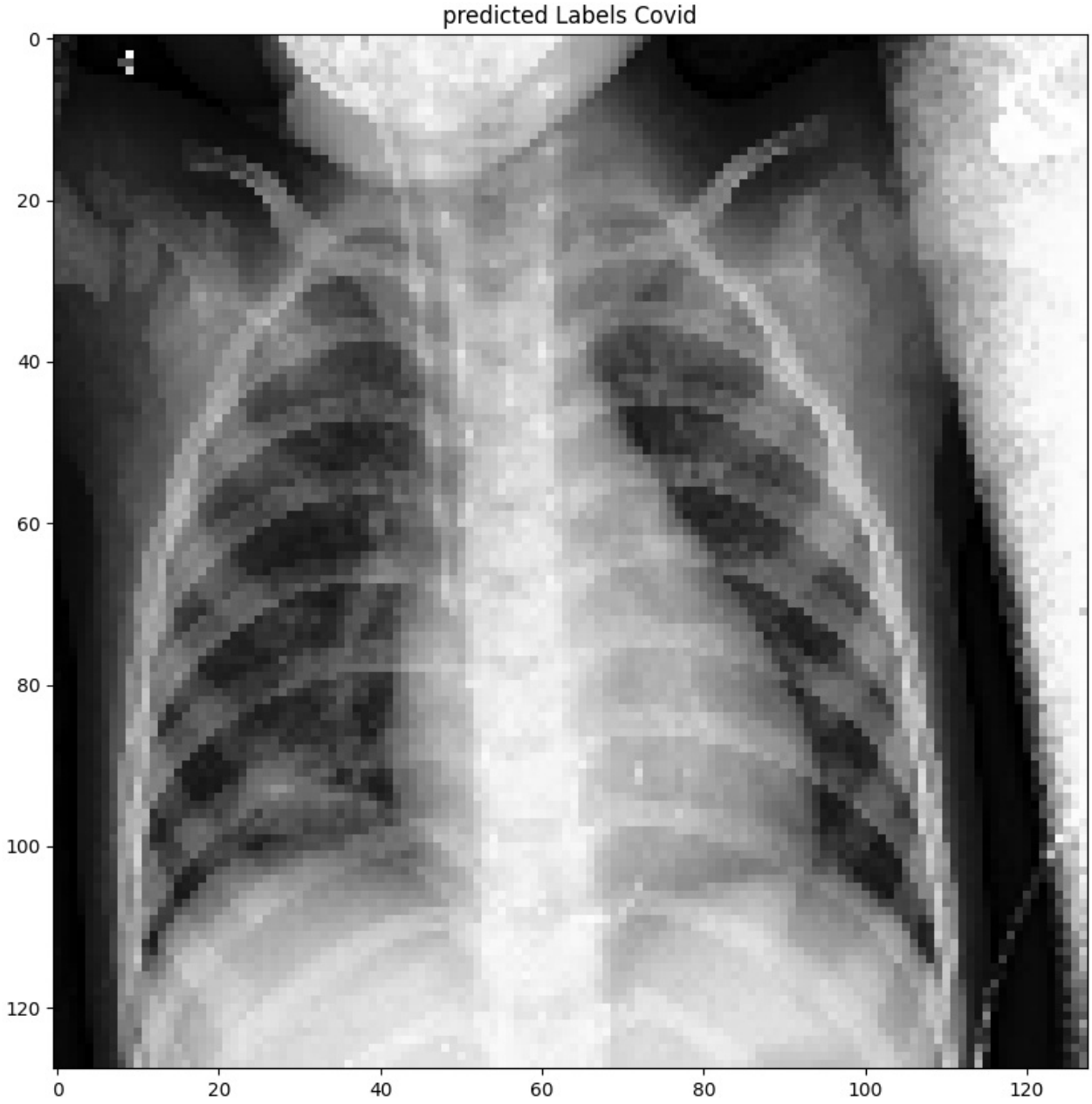

predicted Labels Viral Pneumonia
0
20
40
60
80
100
120
0
20
40
60
80
100
120

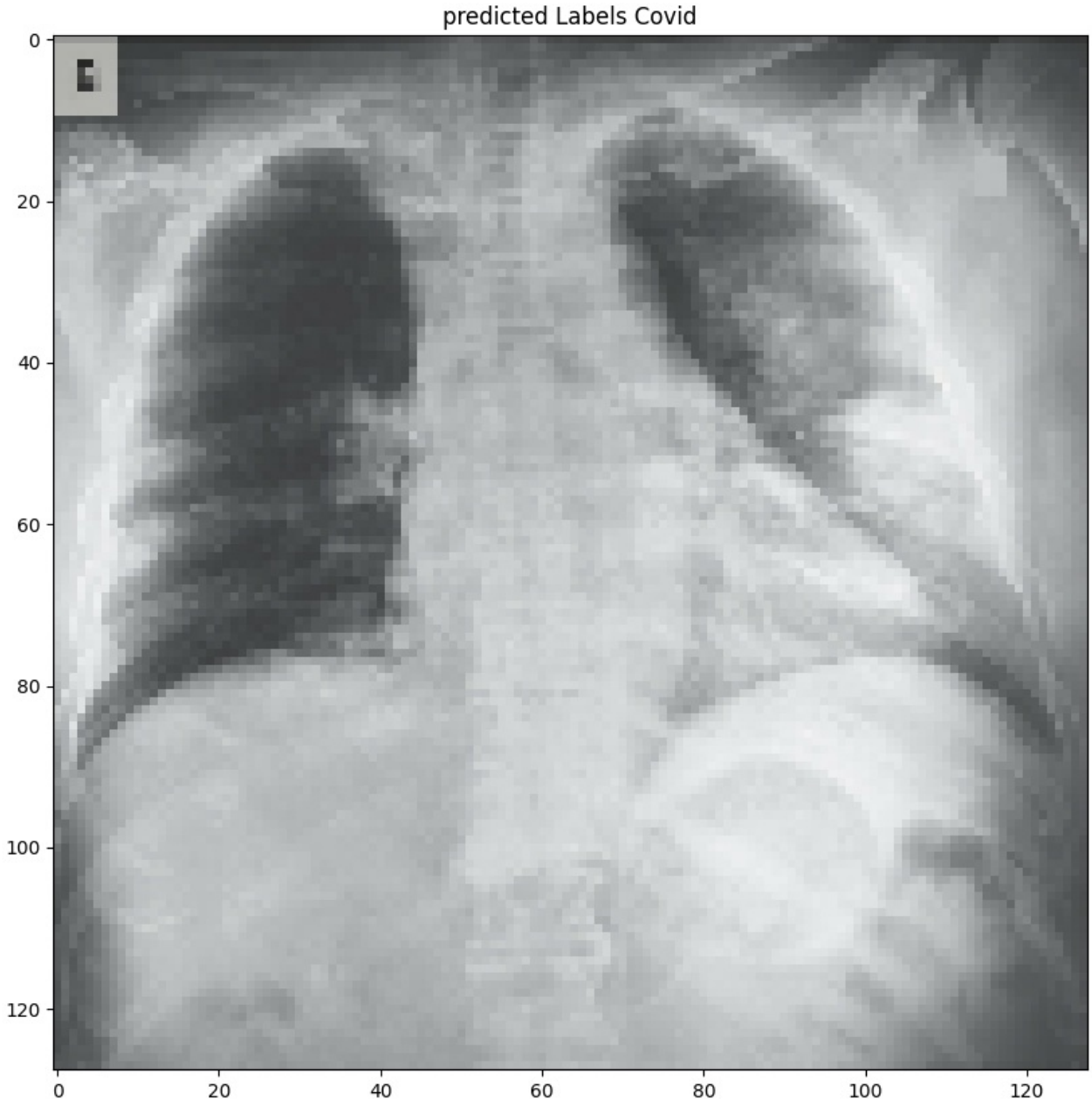

predicted Labels Viral Pneumonia
0
20
40
60
80
100
120
0
20
40
60
80
100
120

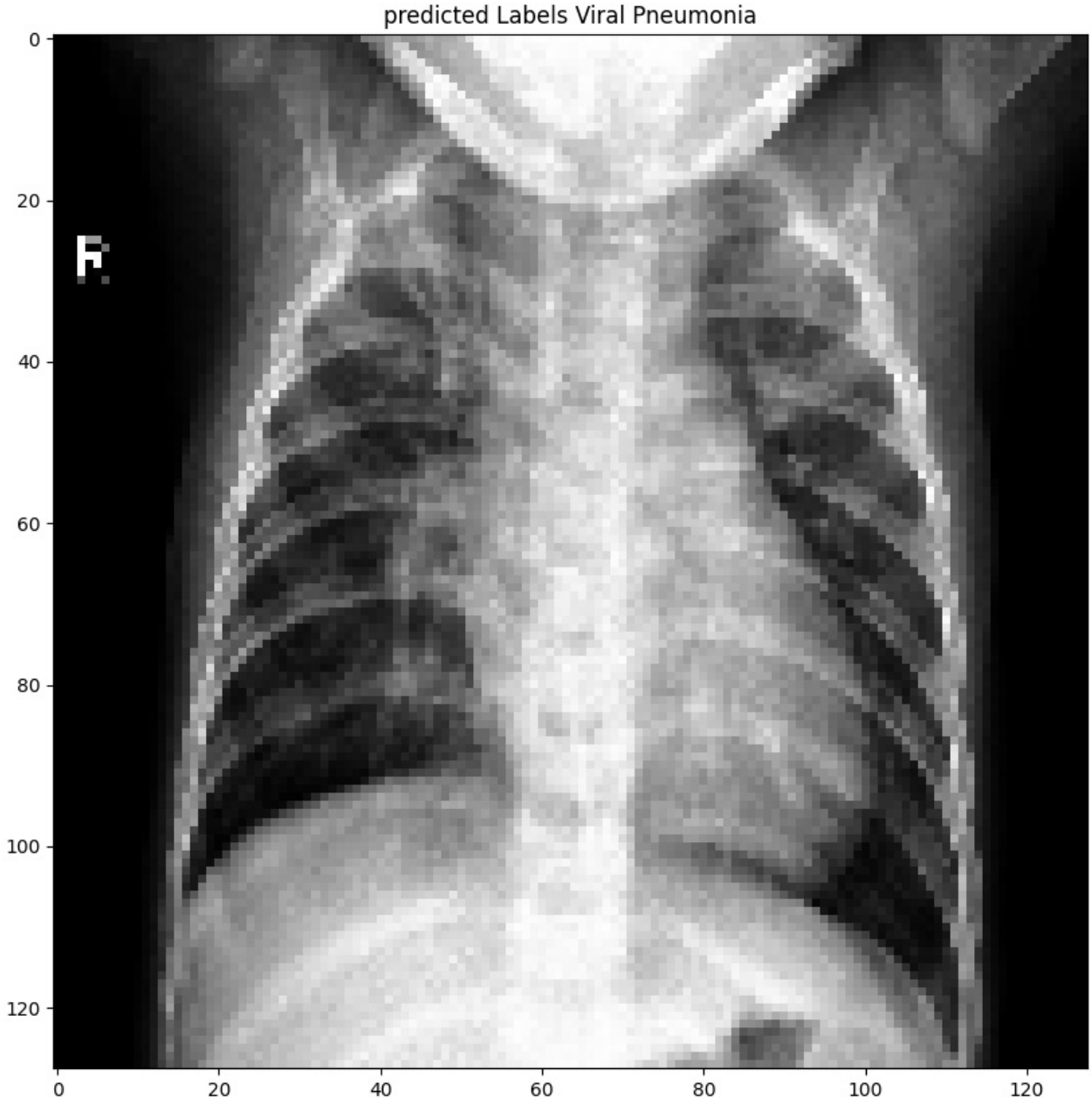

predicted Labels Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

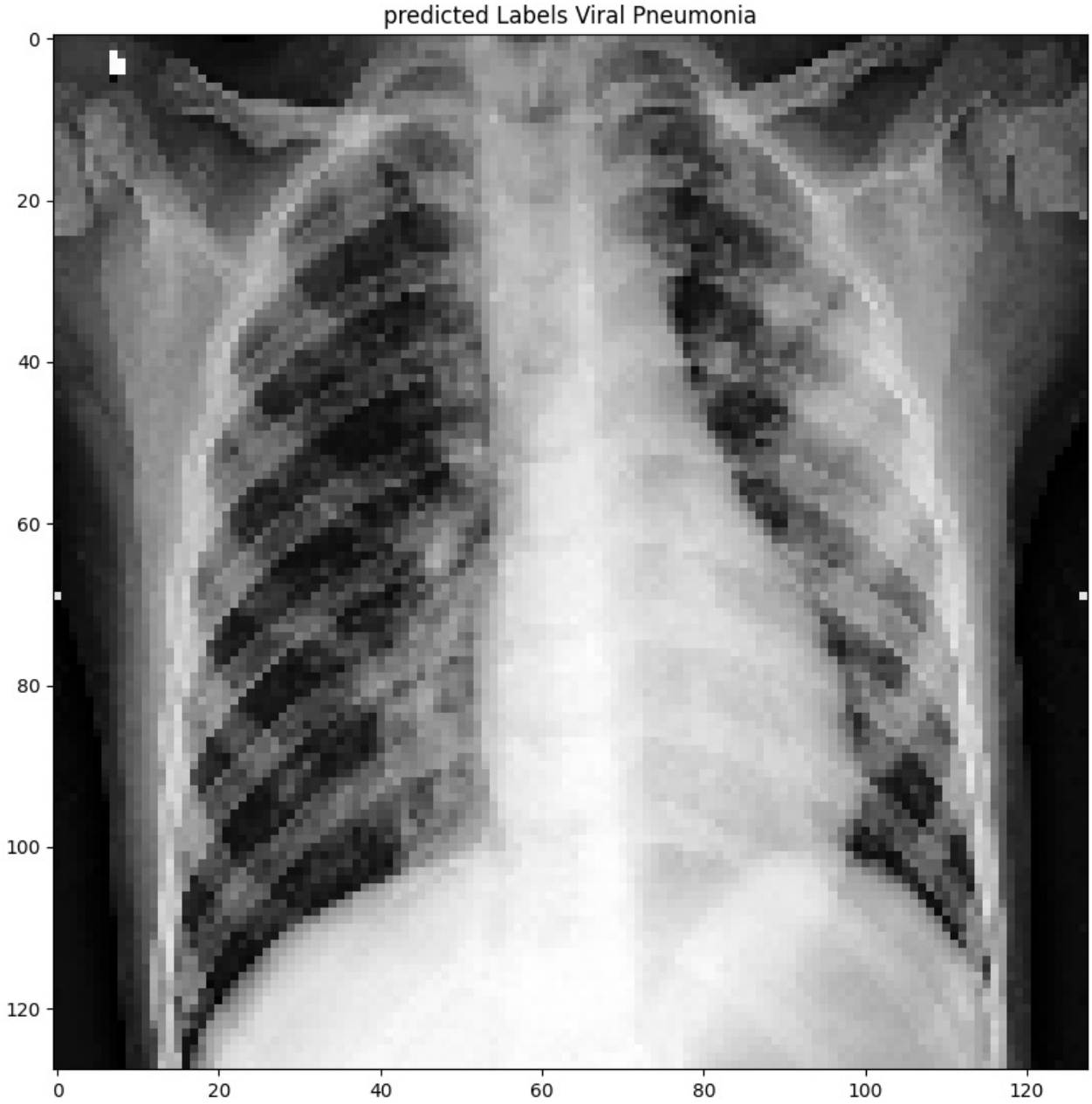

predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

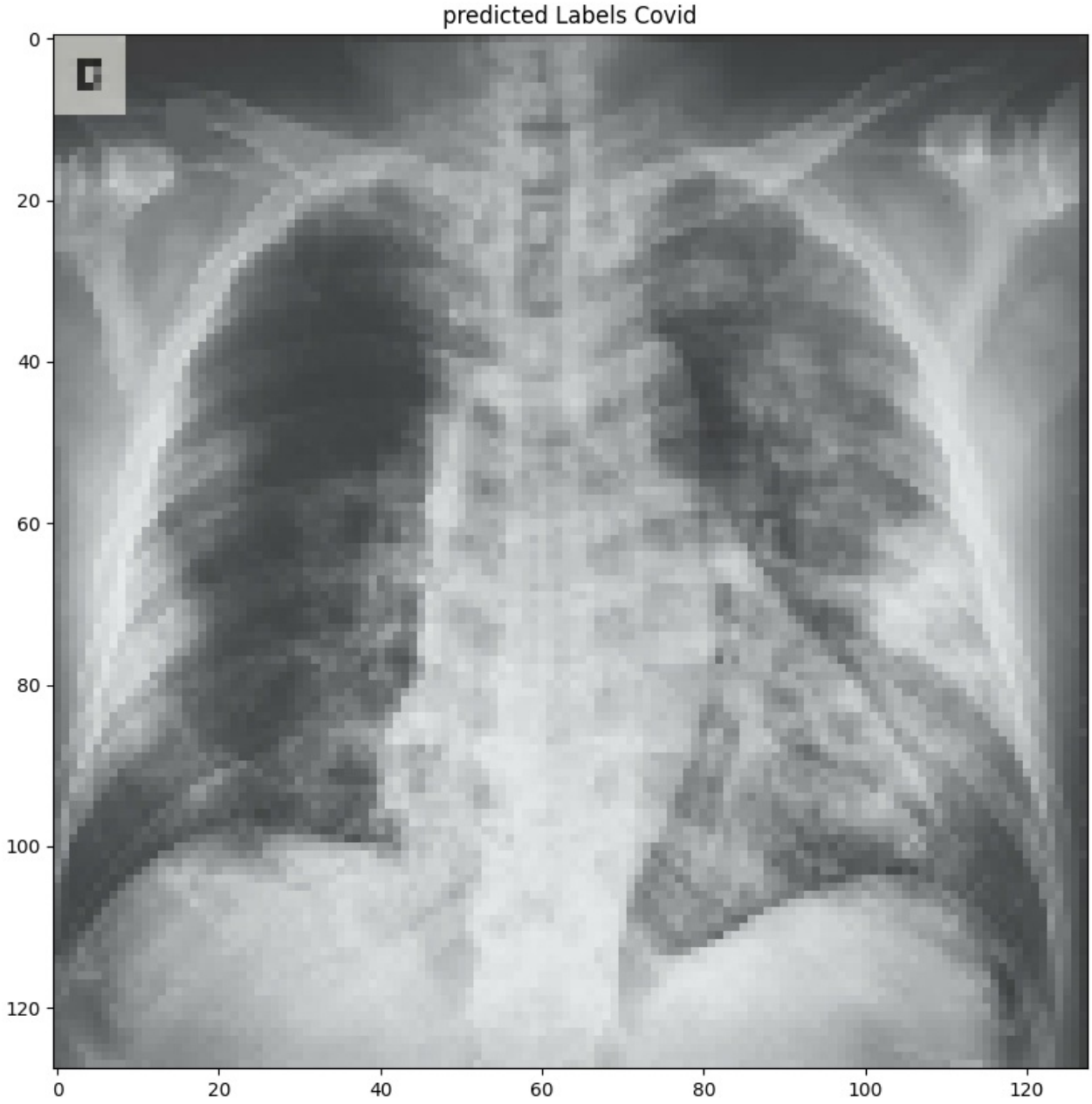

predicted Labels Normal
R
0
20
40
60
80
100
120
0
20
40
60
80
100
120

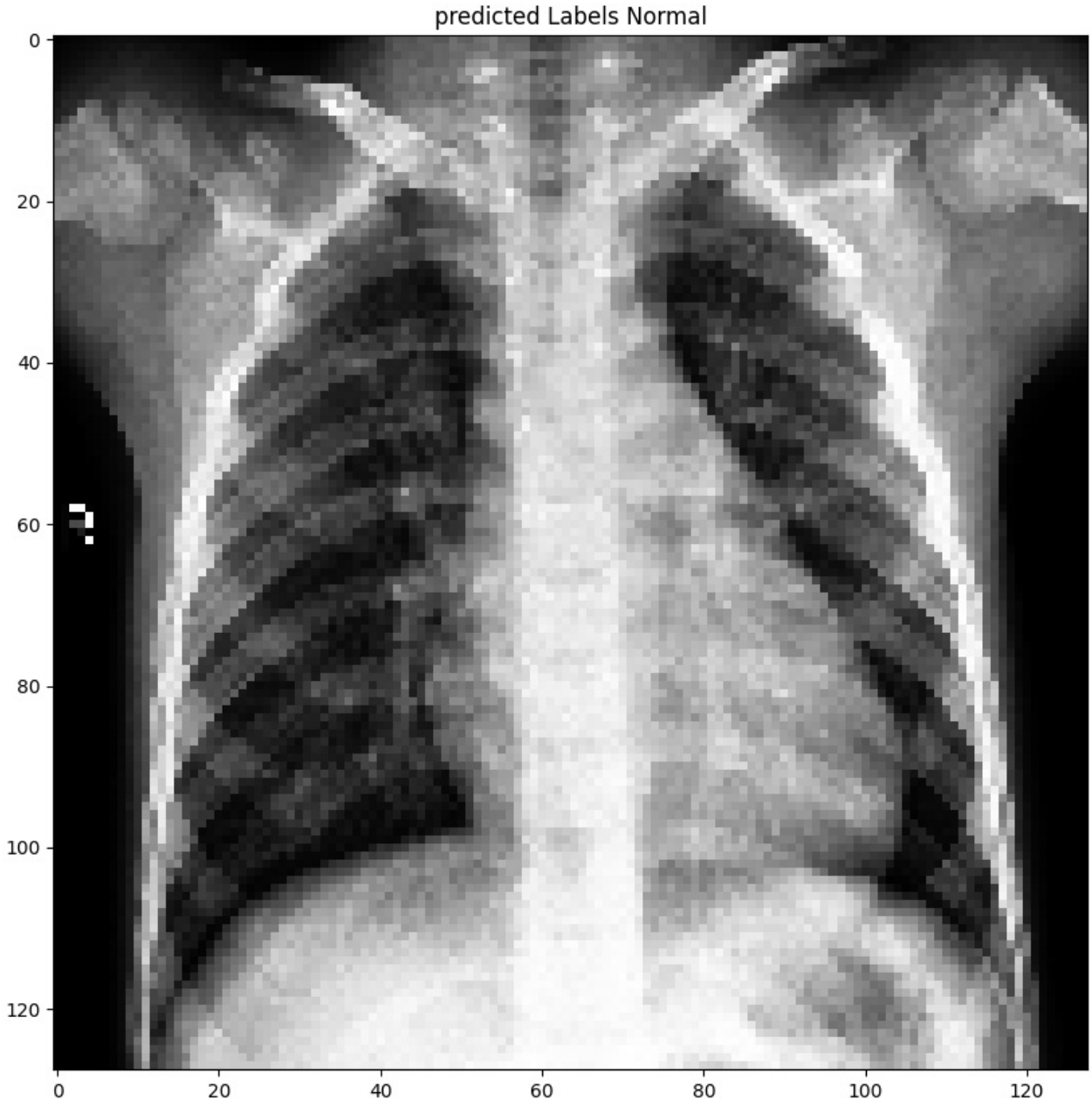

predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

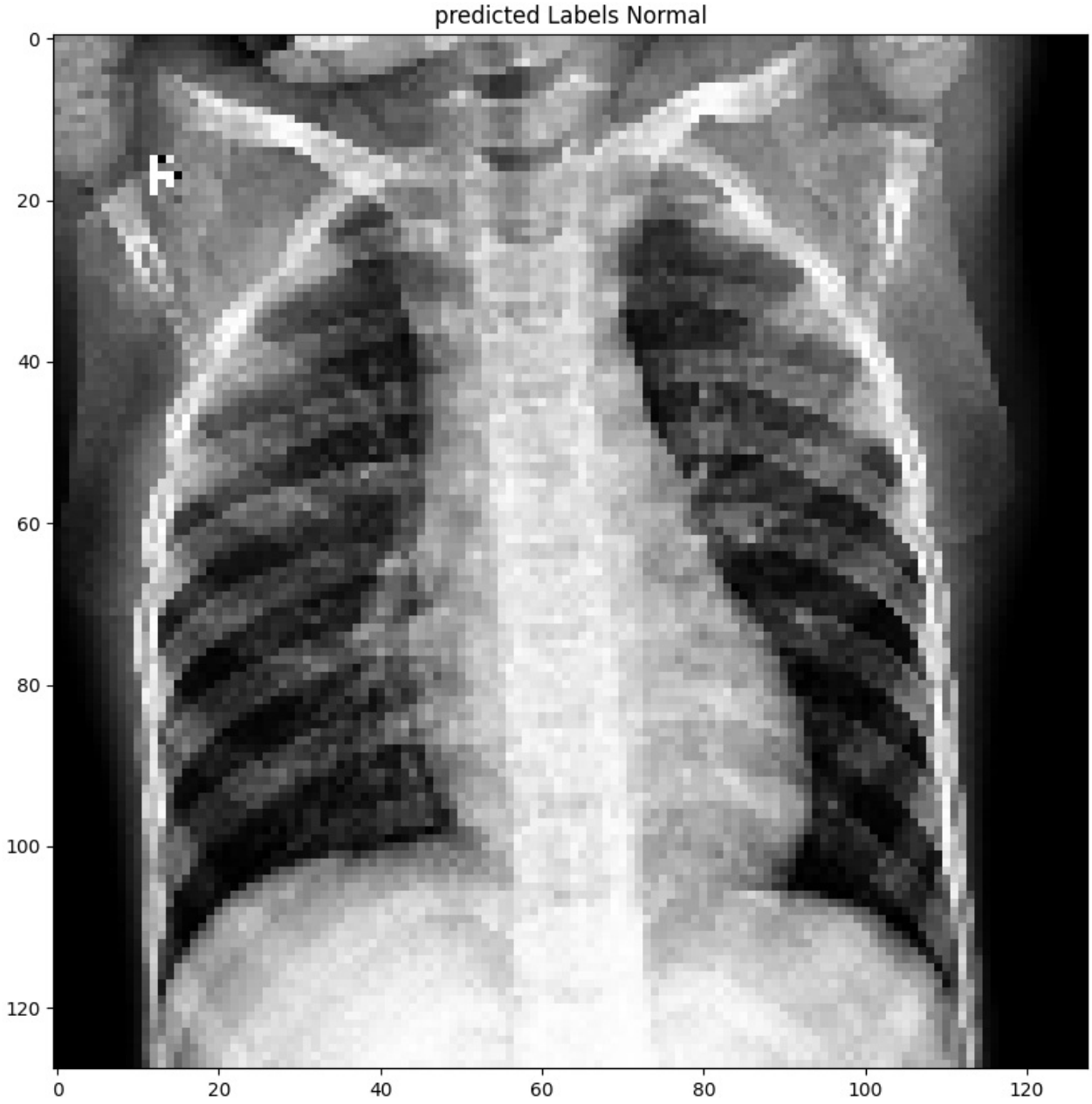

predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

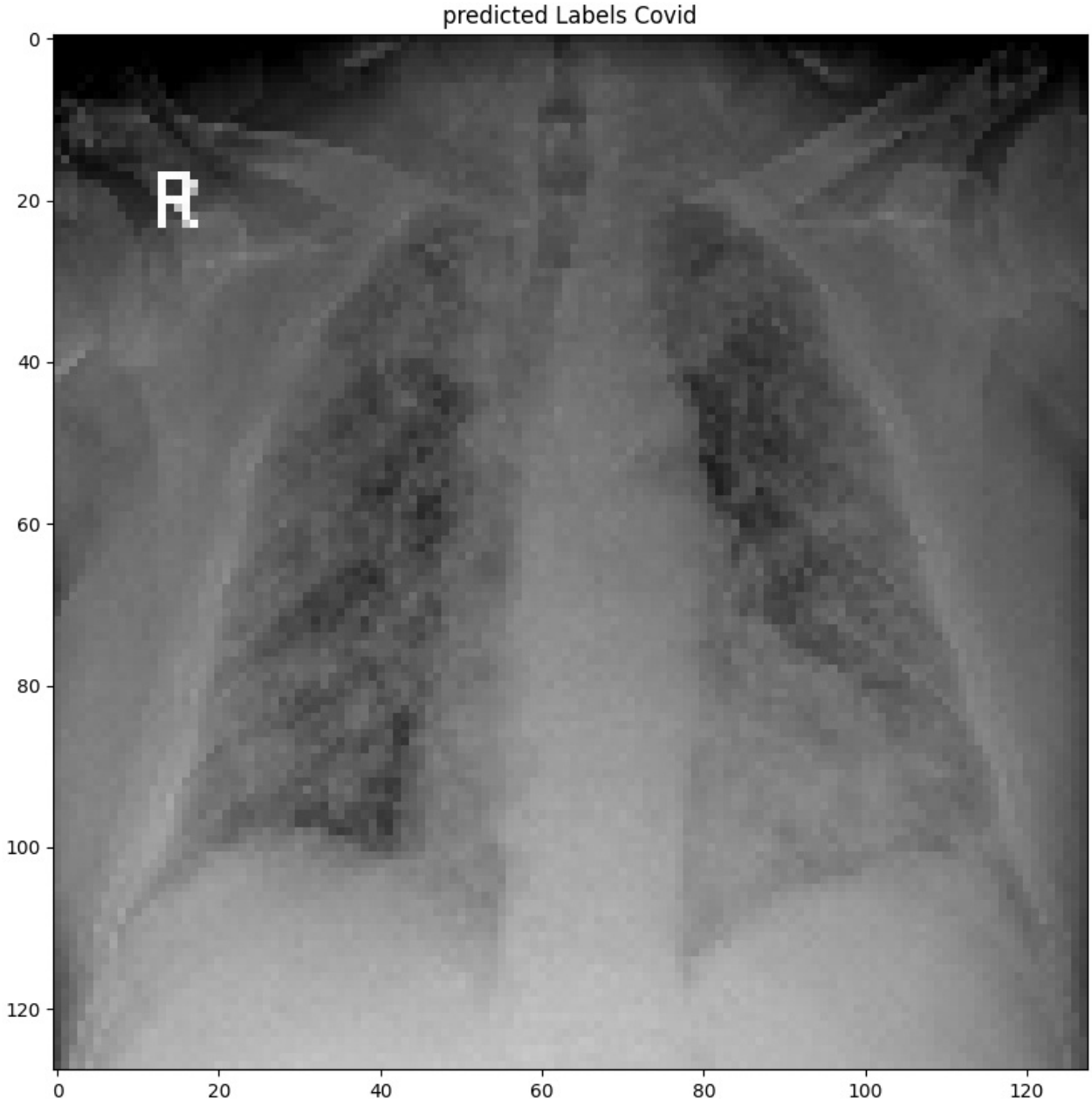

predicted Labels Viral Pneumonia
0
20
40
60
80
100
120
0
20
40
60
80
100
120

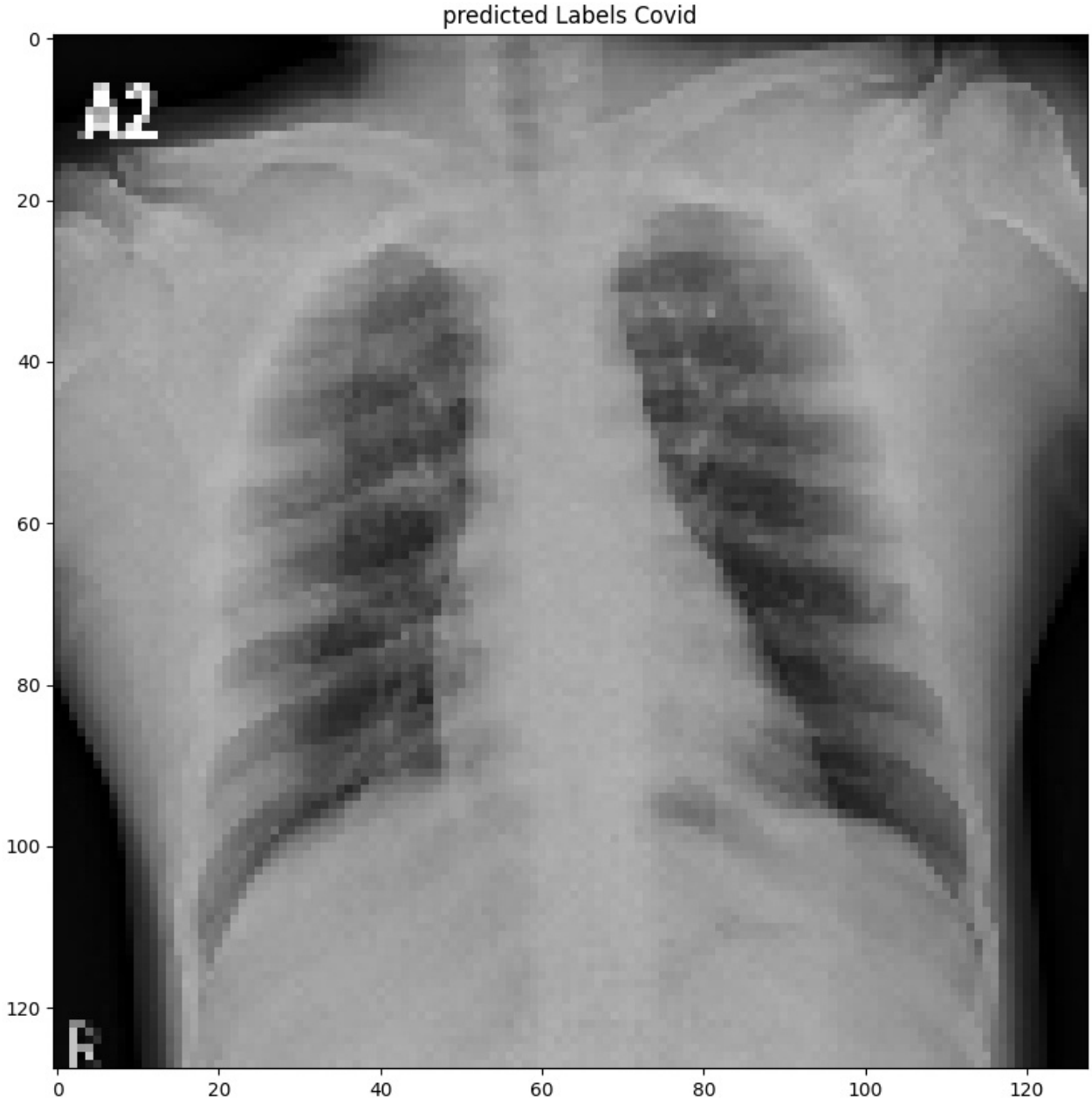

predicted Labels Viral Pneumonia
0
20
40
60
80
100
120
0
20
40
60
80
100
120

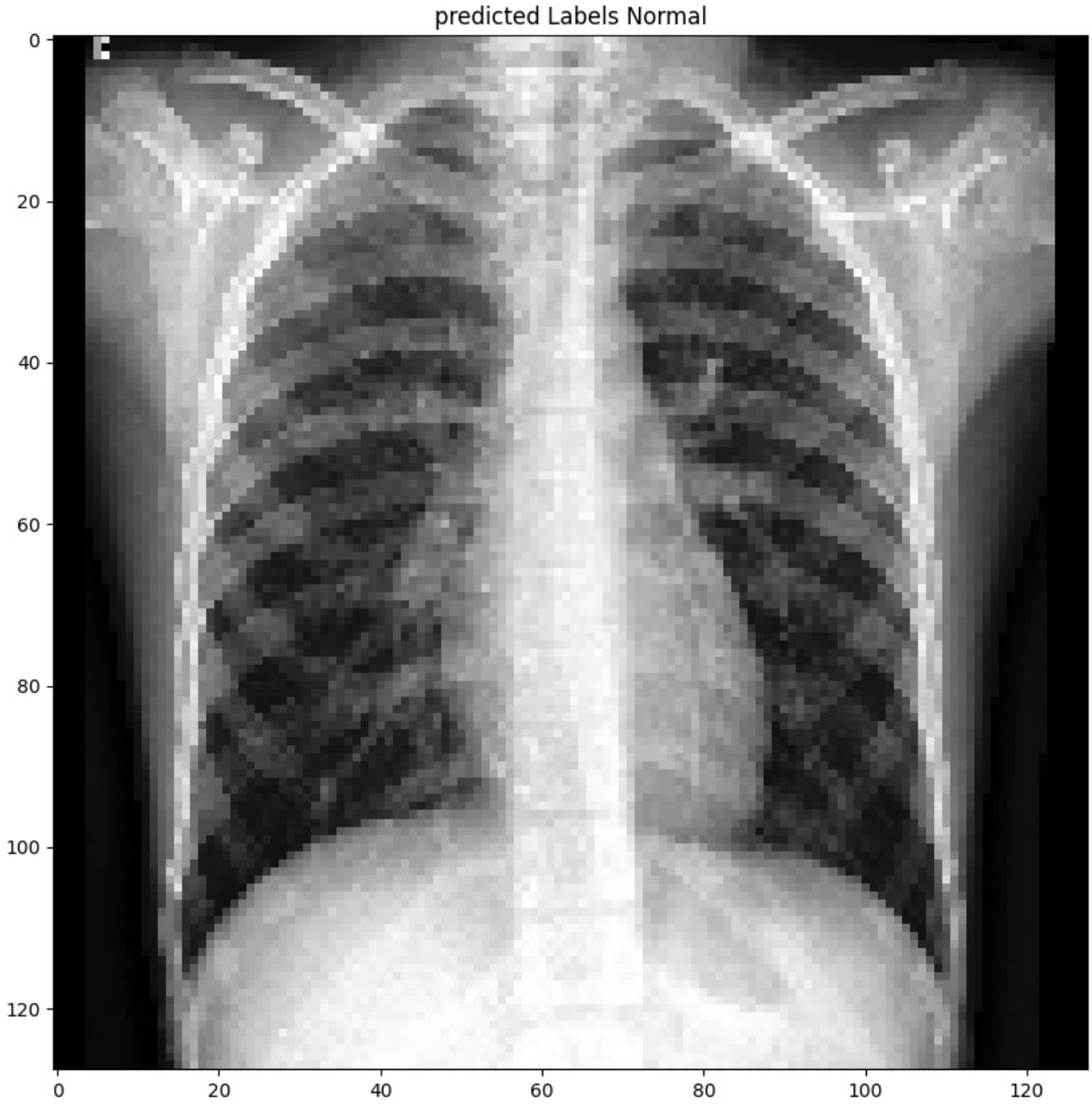

predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

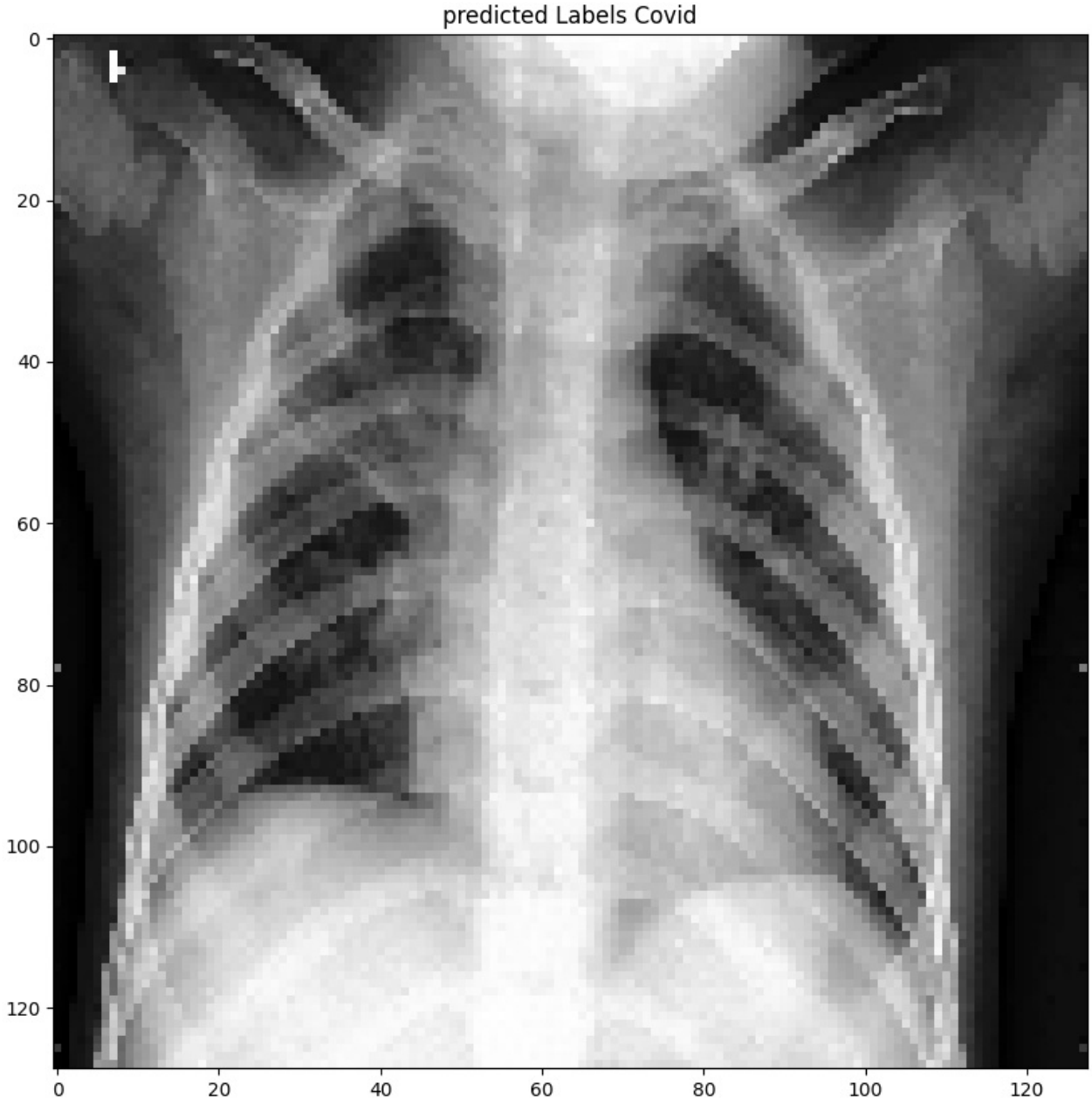

predicted Labels Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

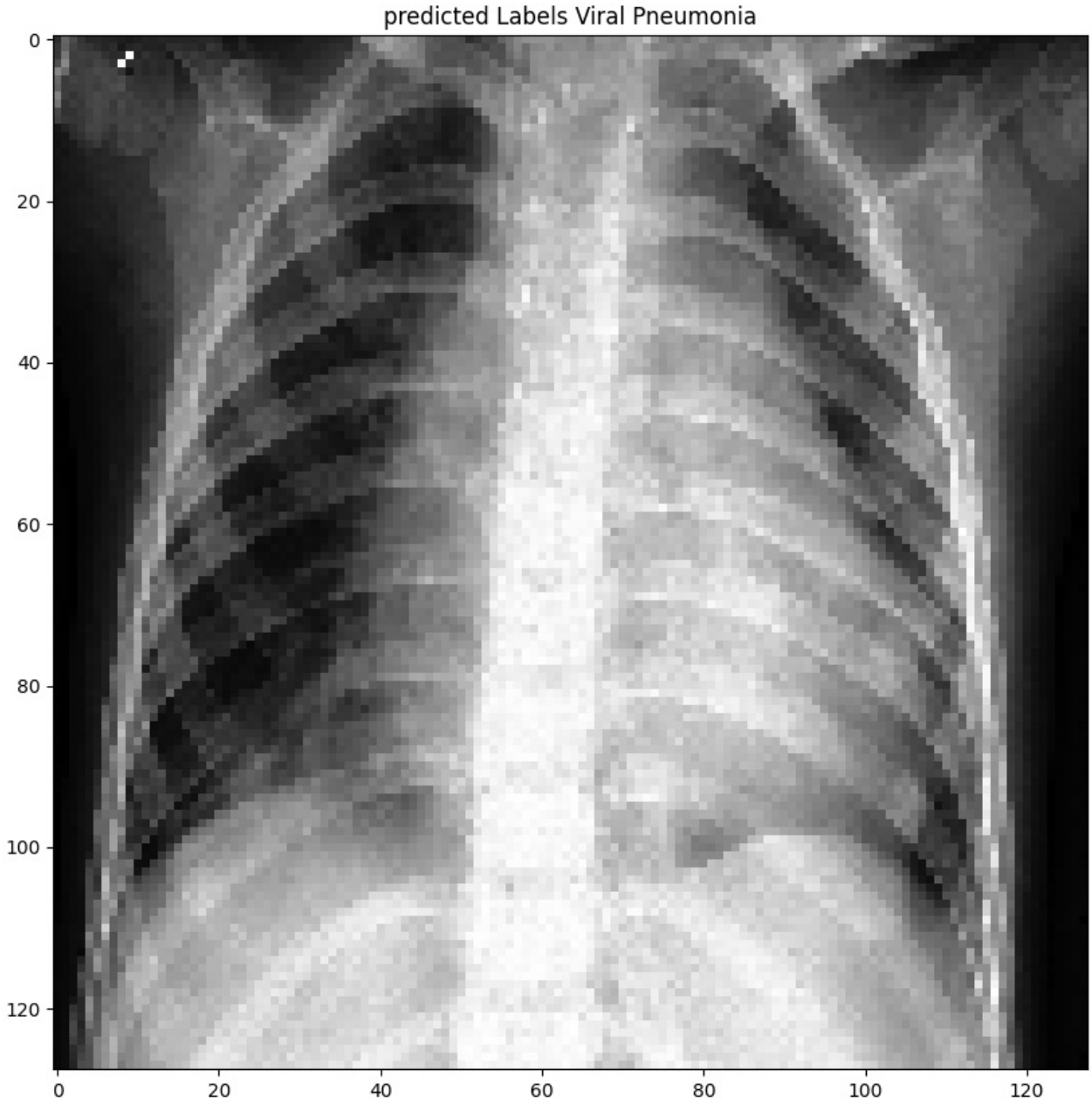

predicted Labels Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

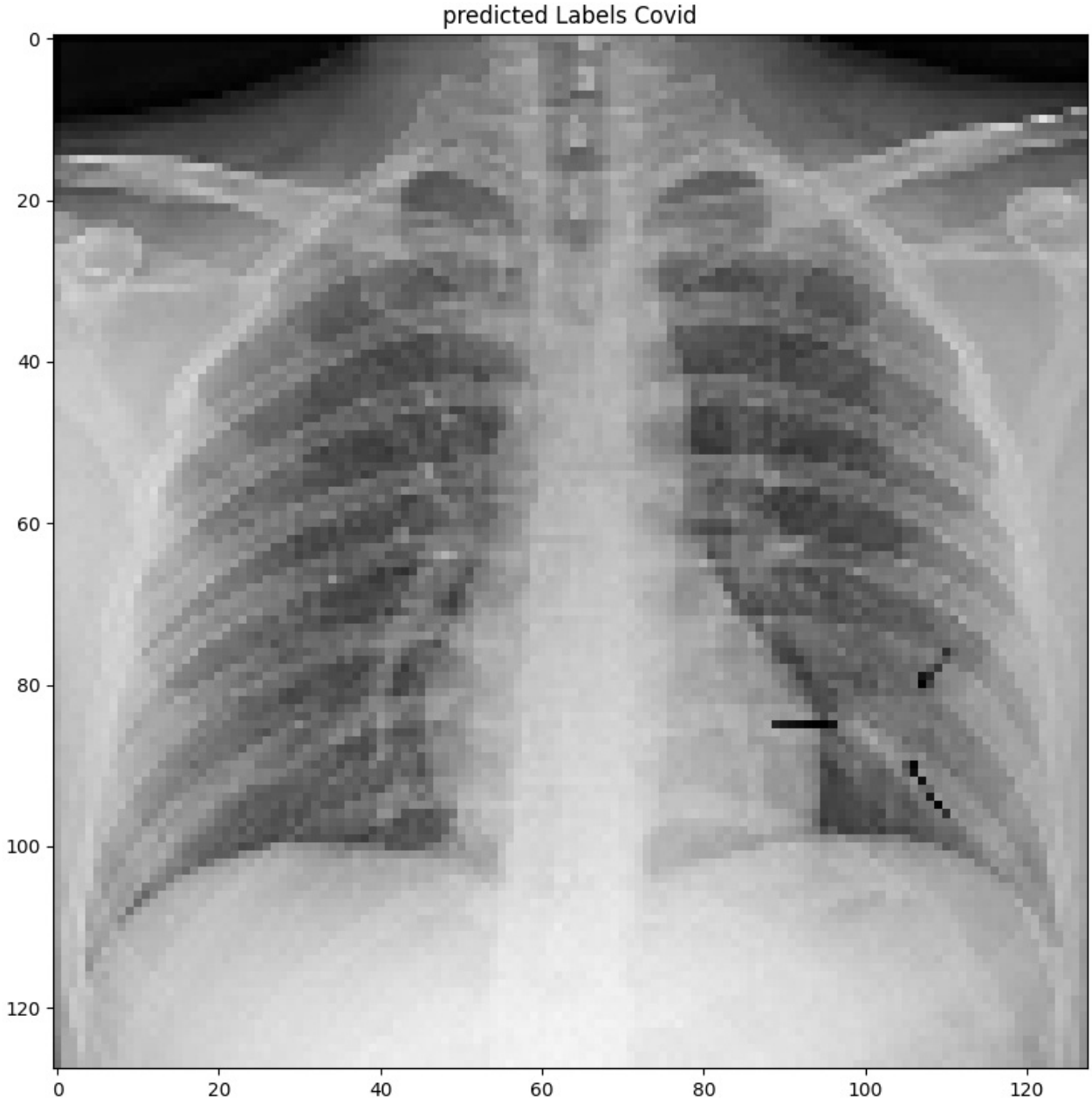

predicted Labels Covid
R
0
20
40
60
80
100
120
0
20
40
60
80
100
120

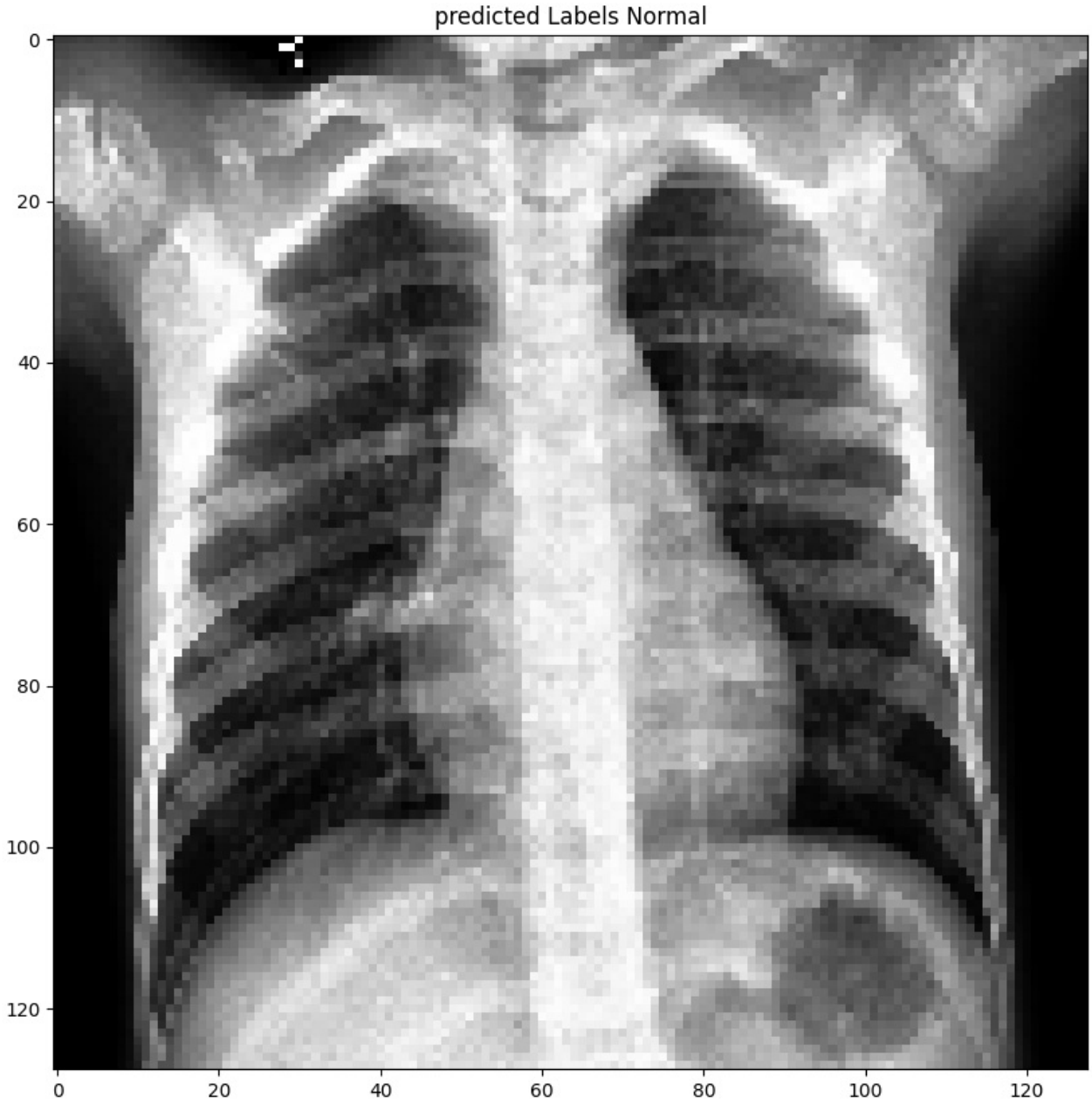

predicted Labels Covid
A2
R
0
20
40
60
80
100
120
0
20
40
60
80
100
120

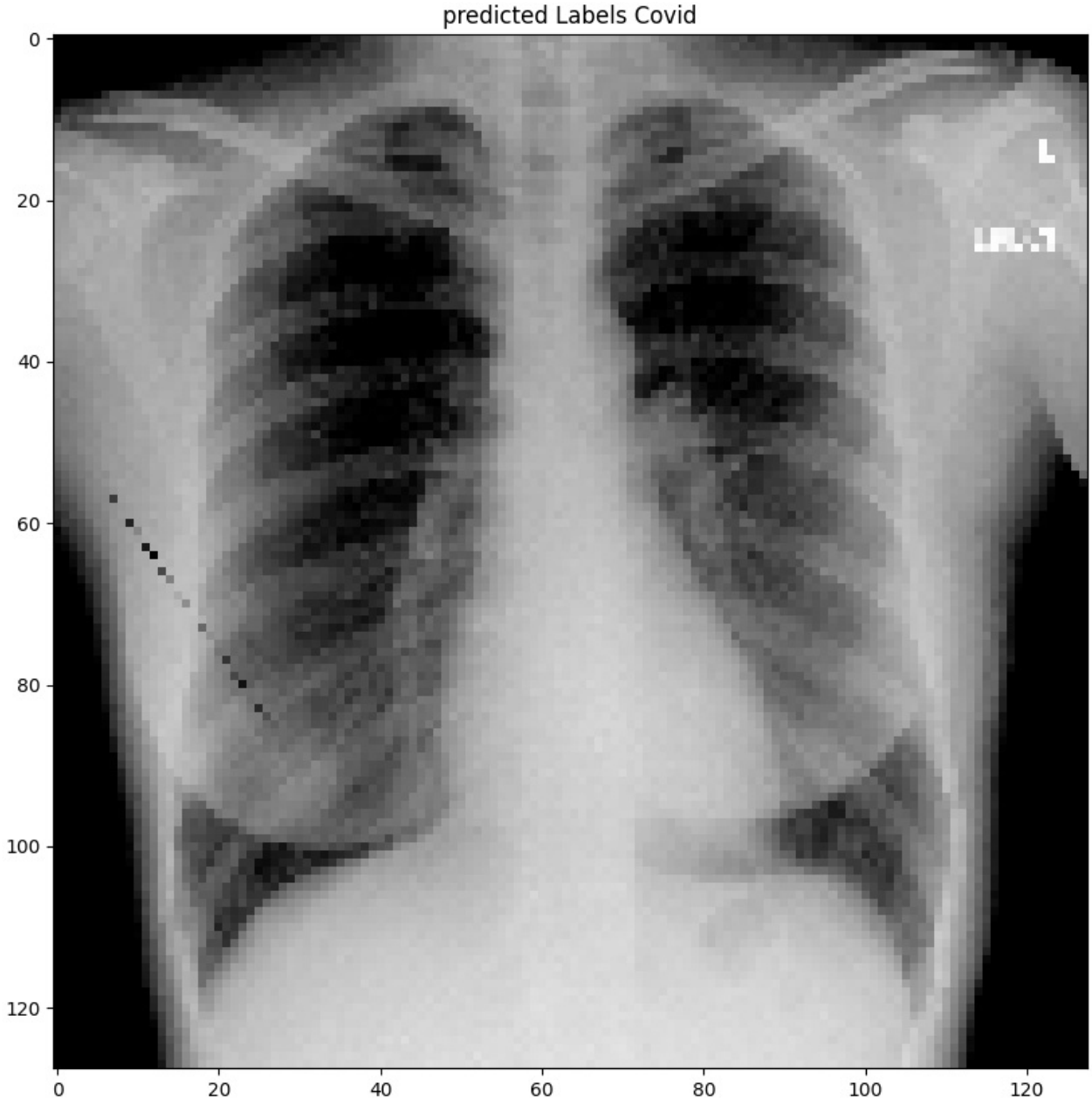

predicted Labels Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

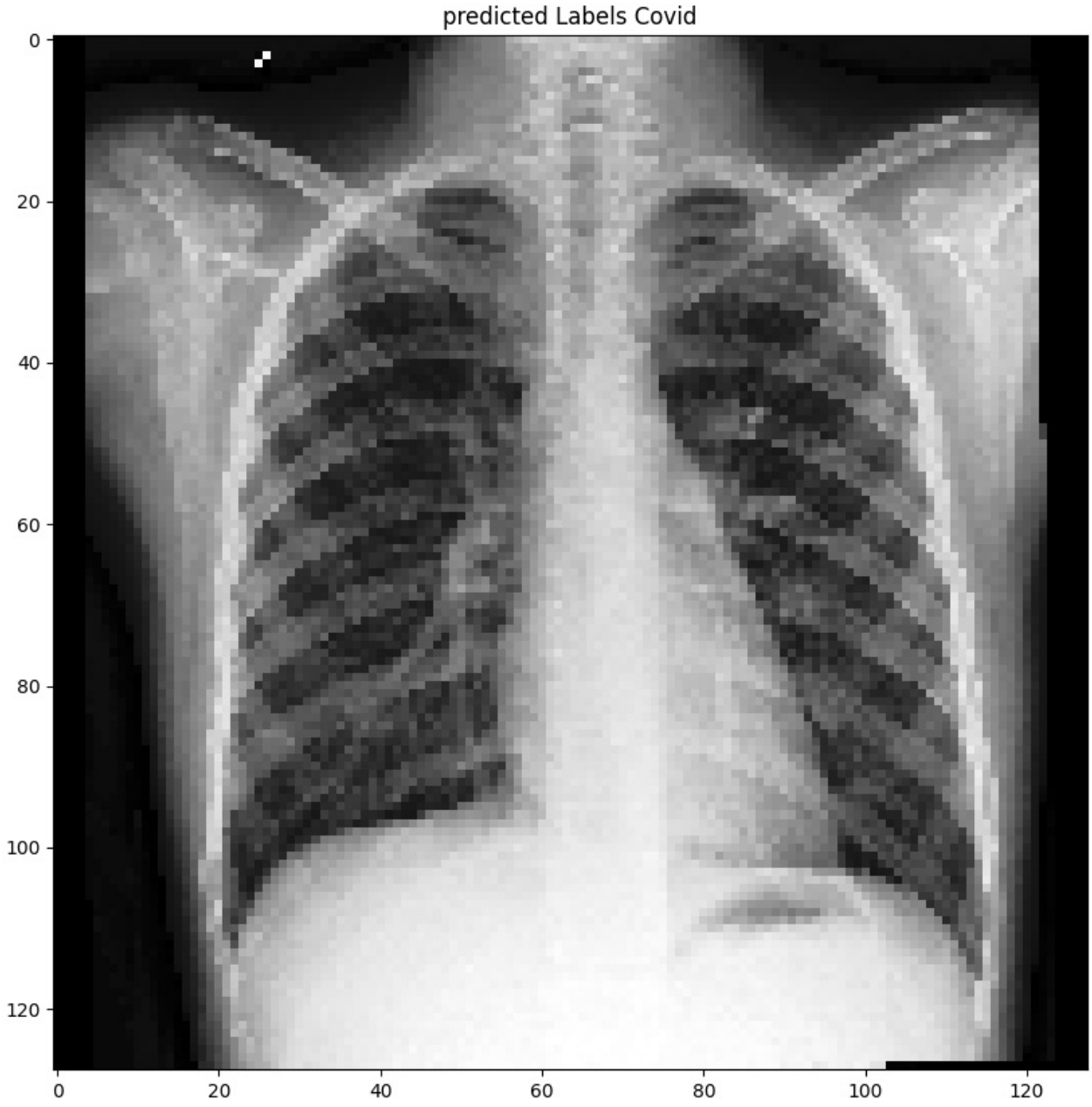

predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

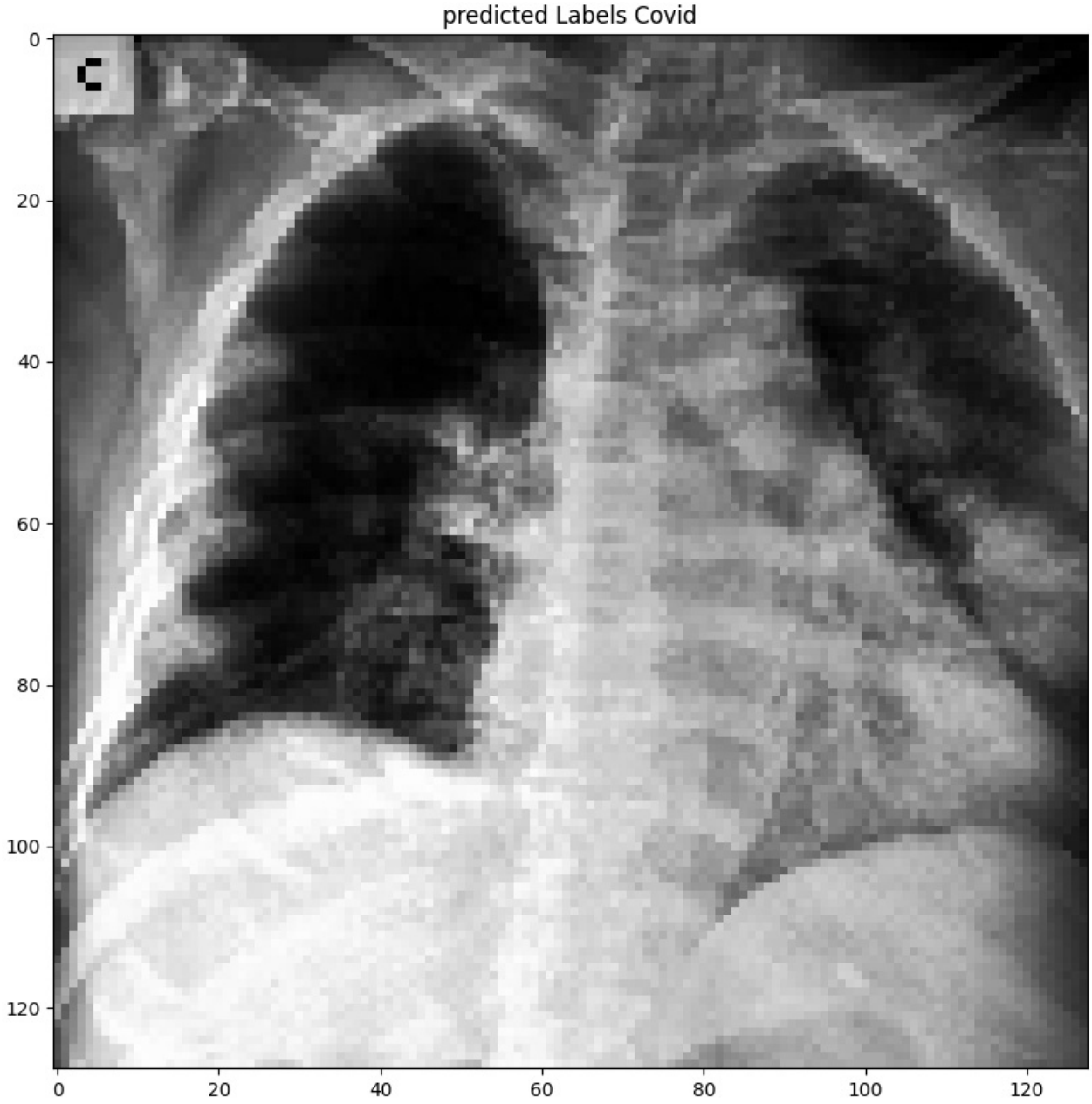

predicted Labels Viral Pneumonia
0
20
40
60
80
100
120
0
20
40
60
80
100
120

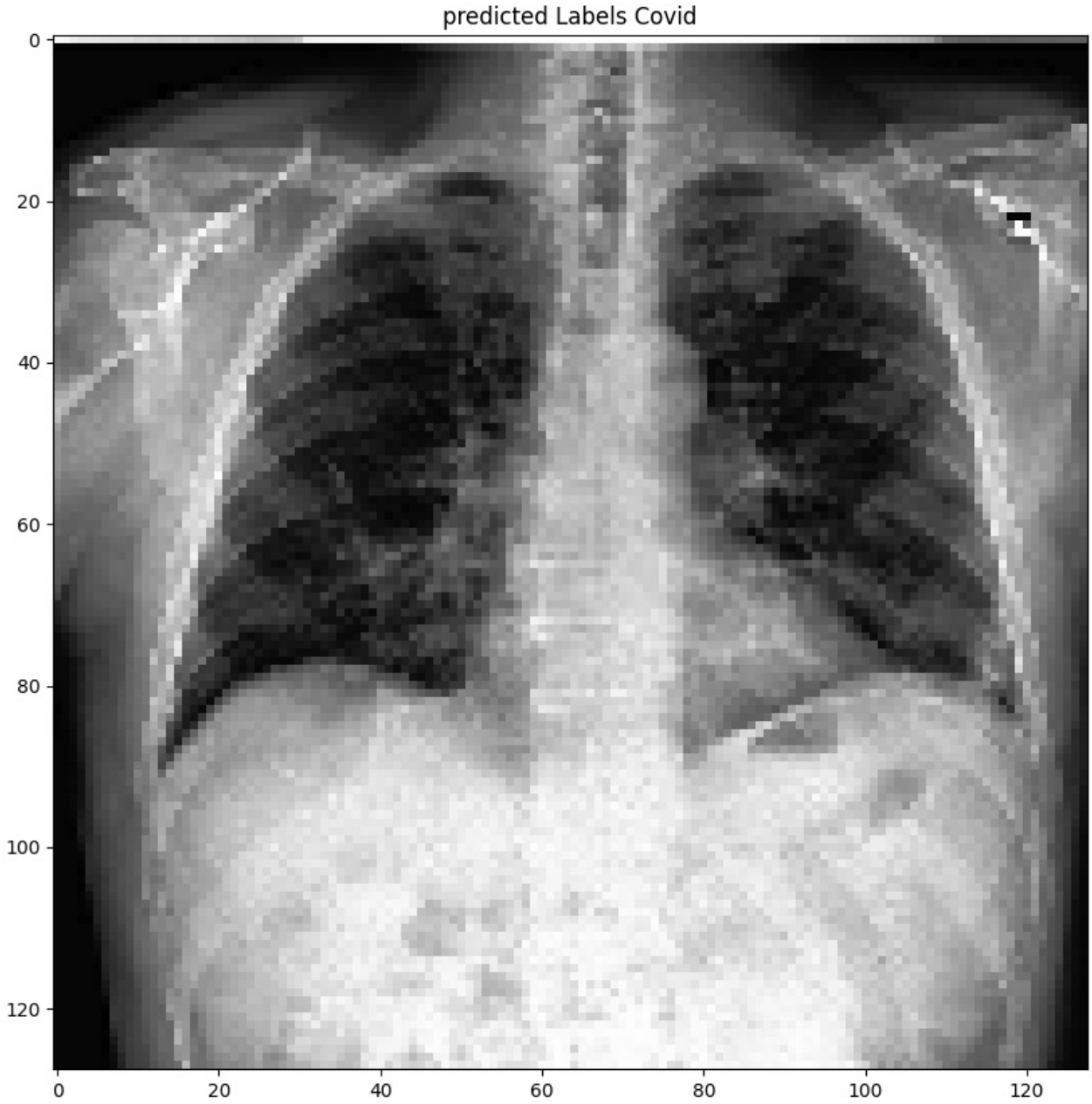

predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

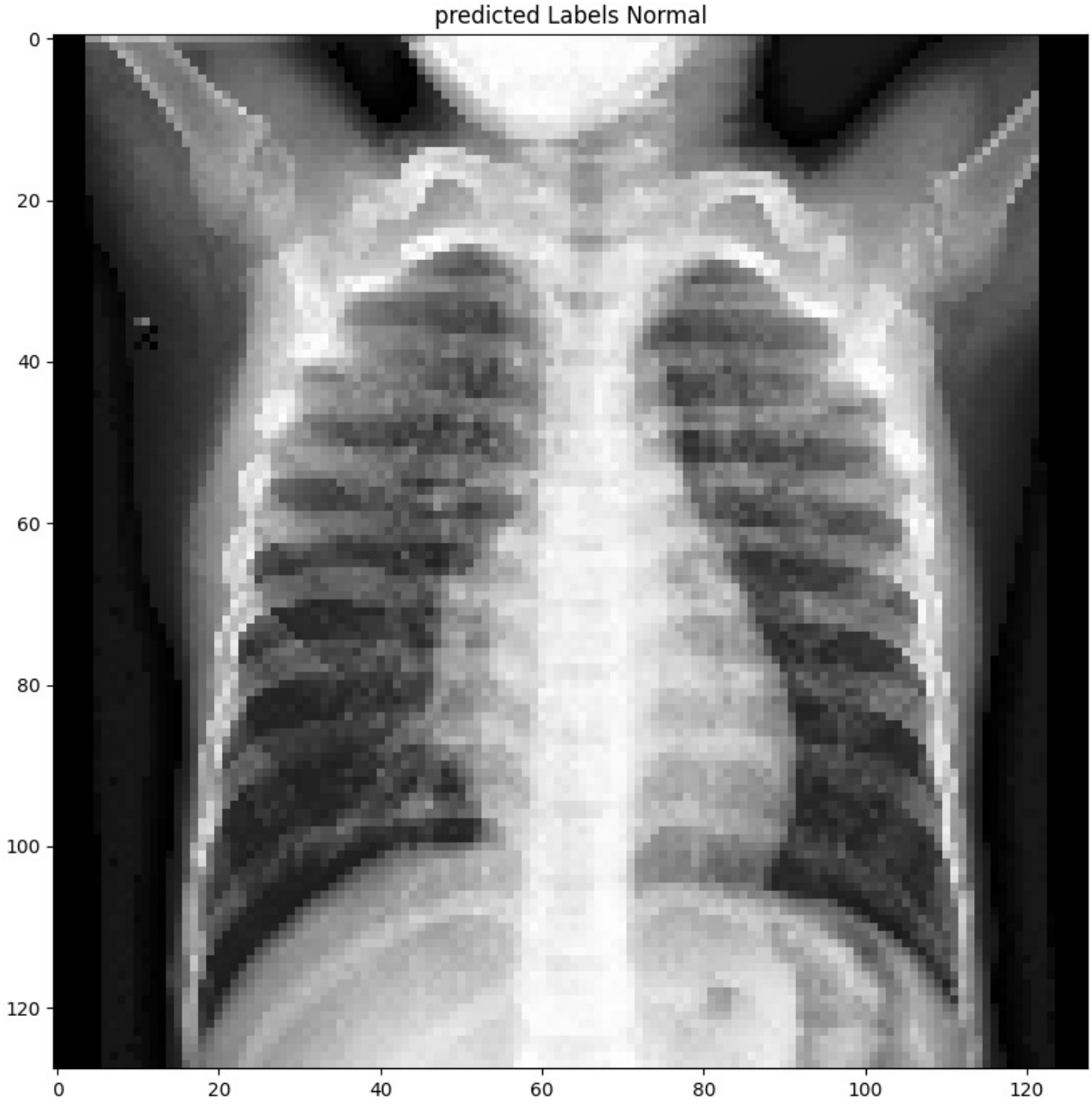

predicted Labels Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

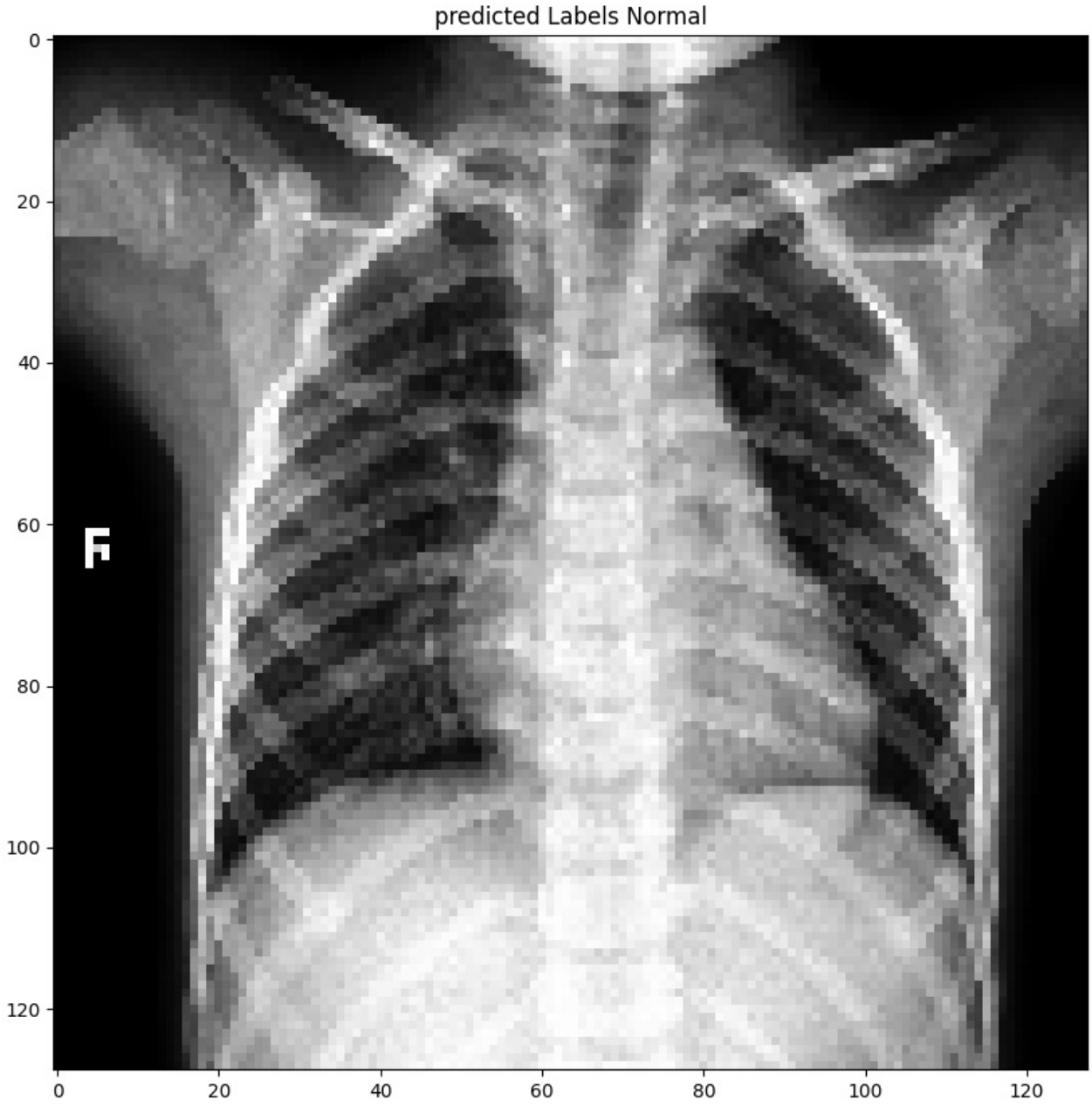

predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120
L

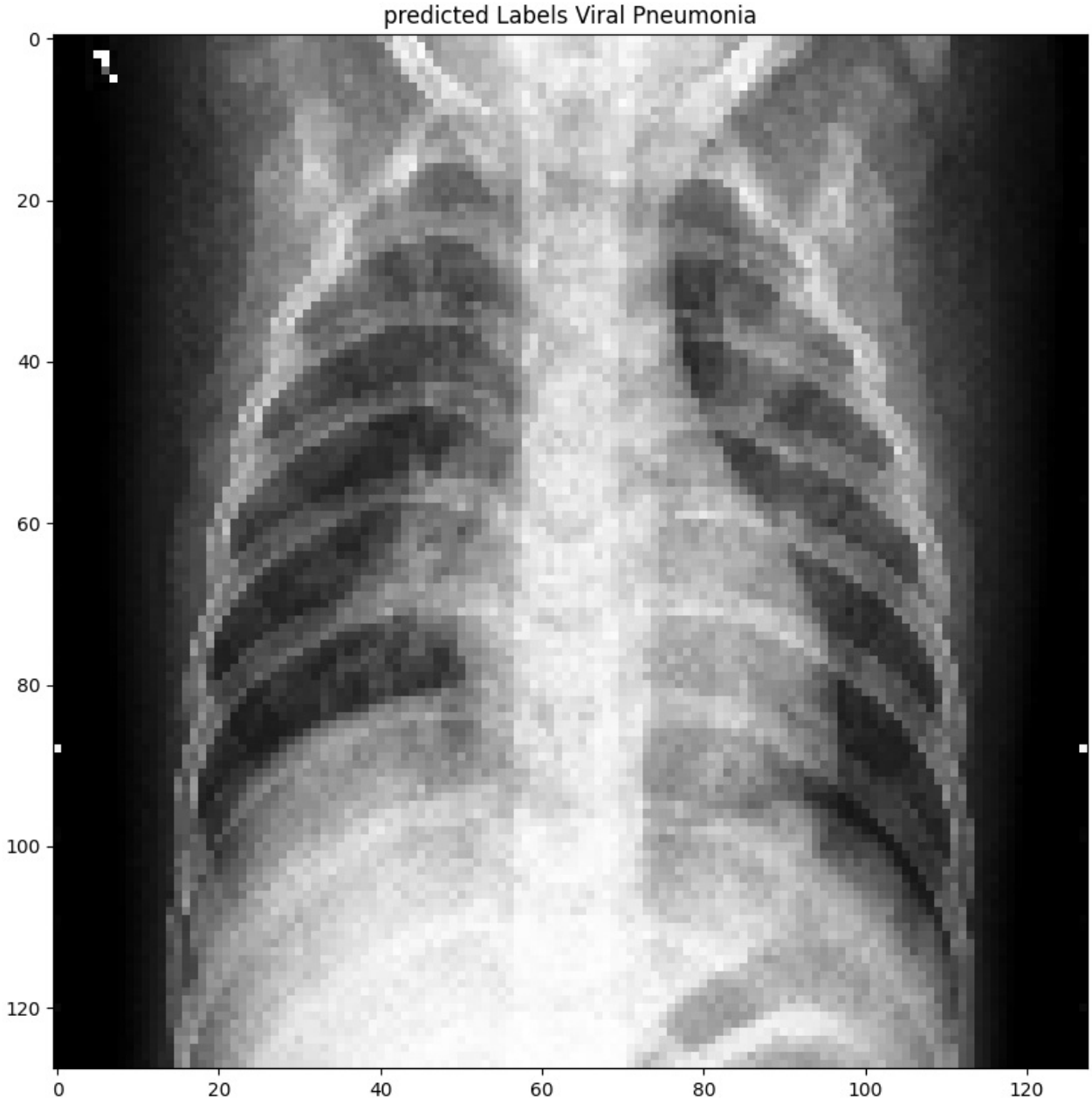

predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

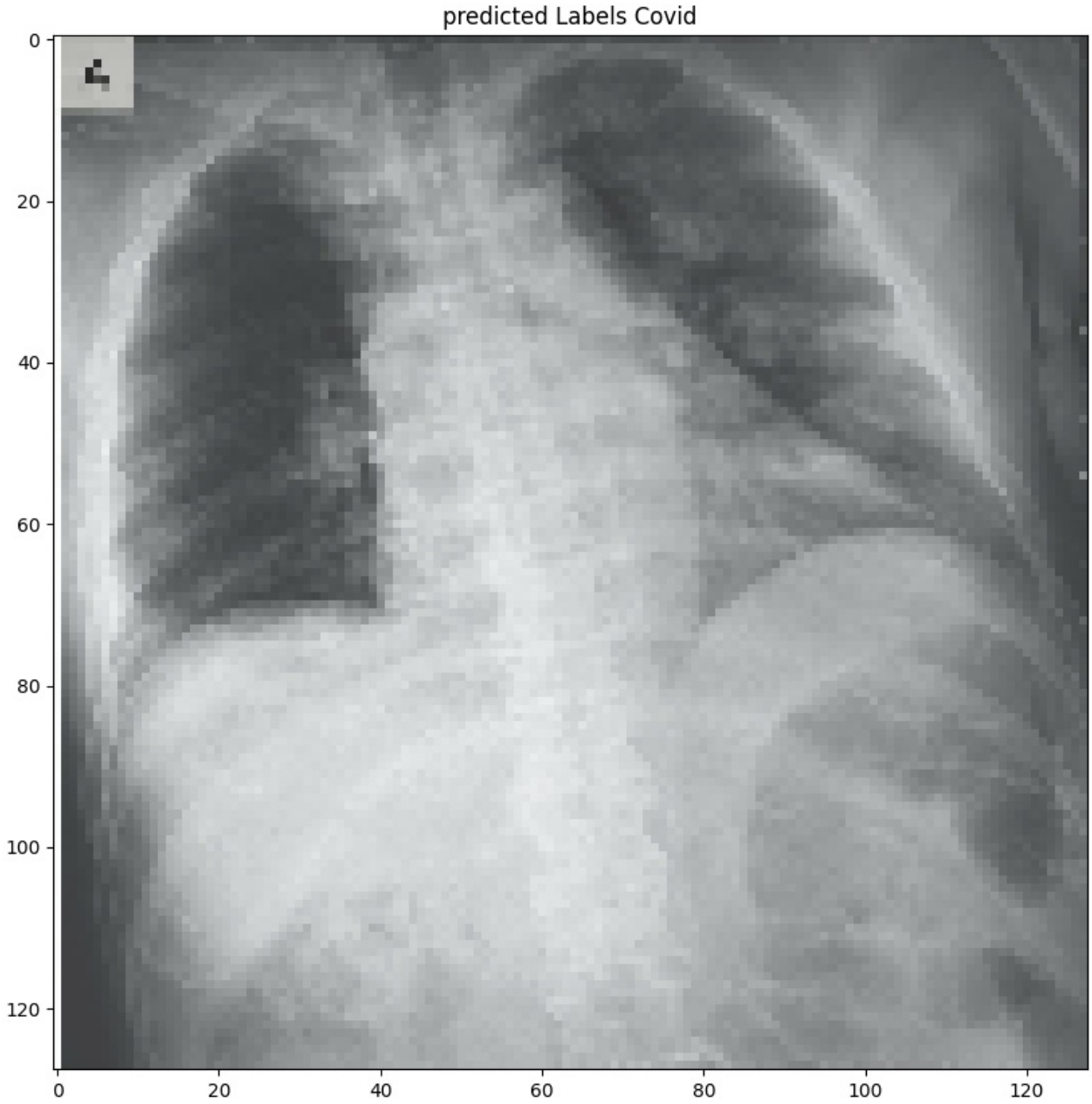

predicted Labels Covid
C
0
20
40
60
80
100
120
0
20
40
60
80
100
120

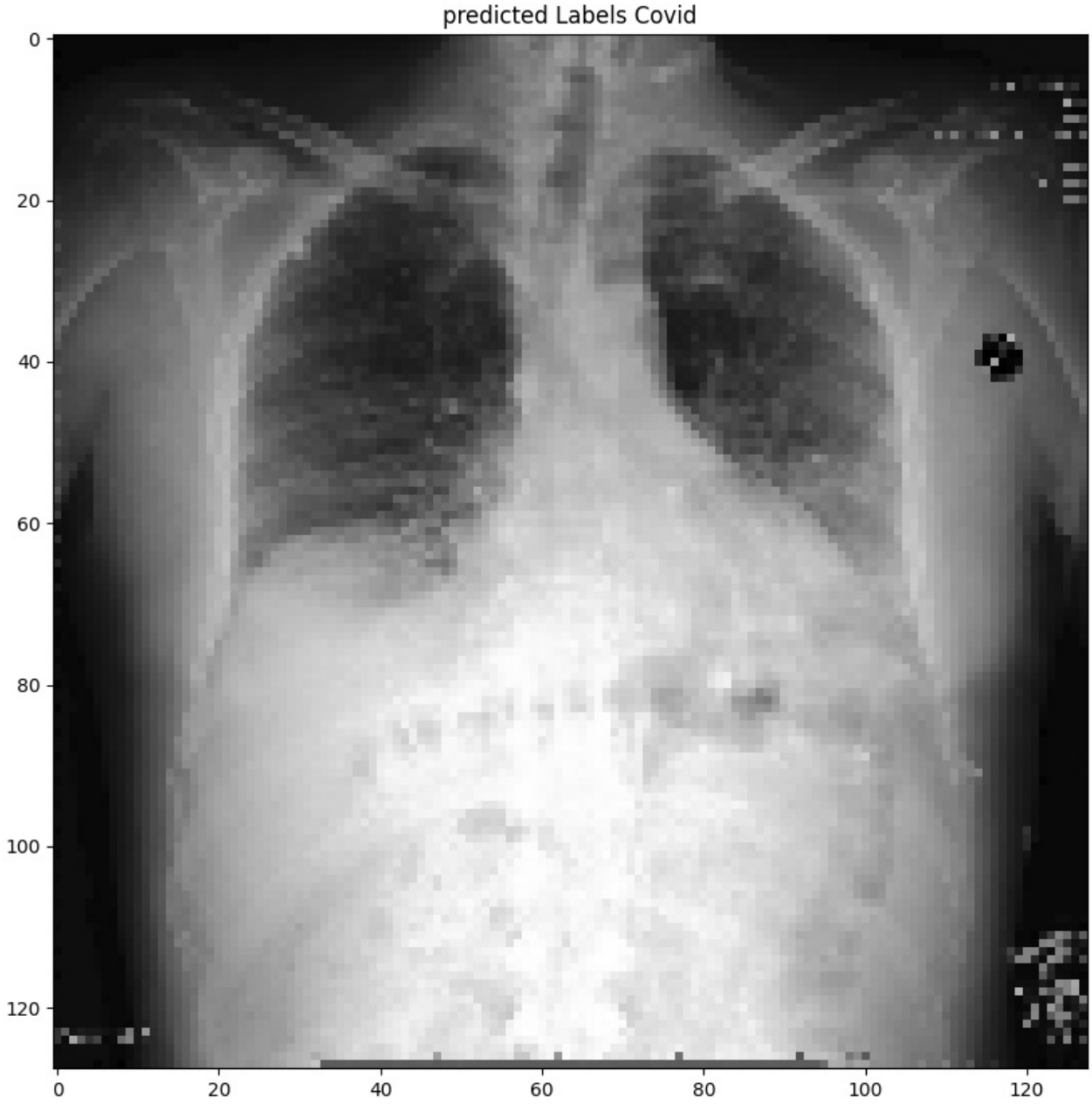

predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

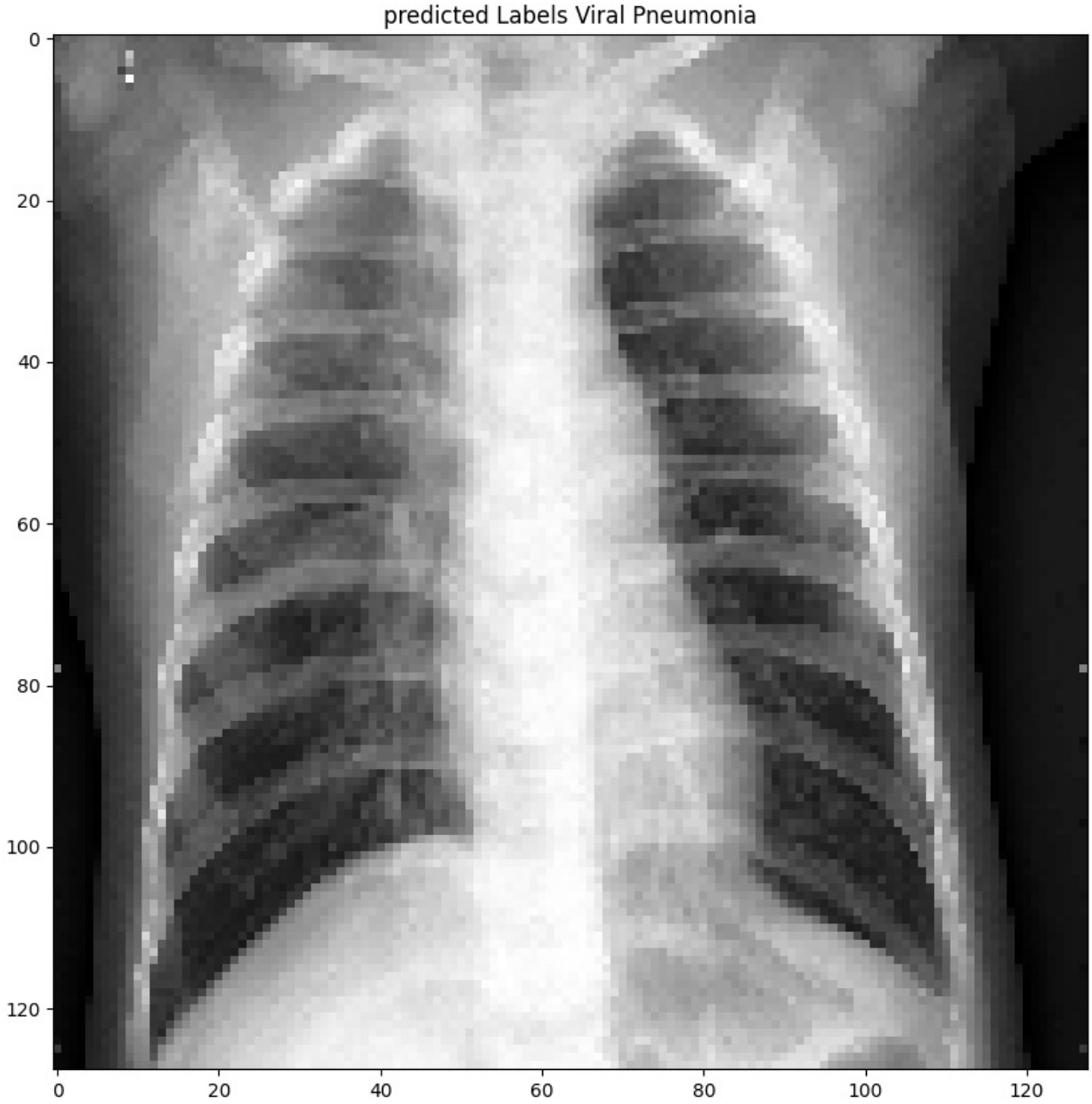

predicted Labels Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

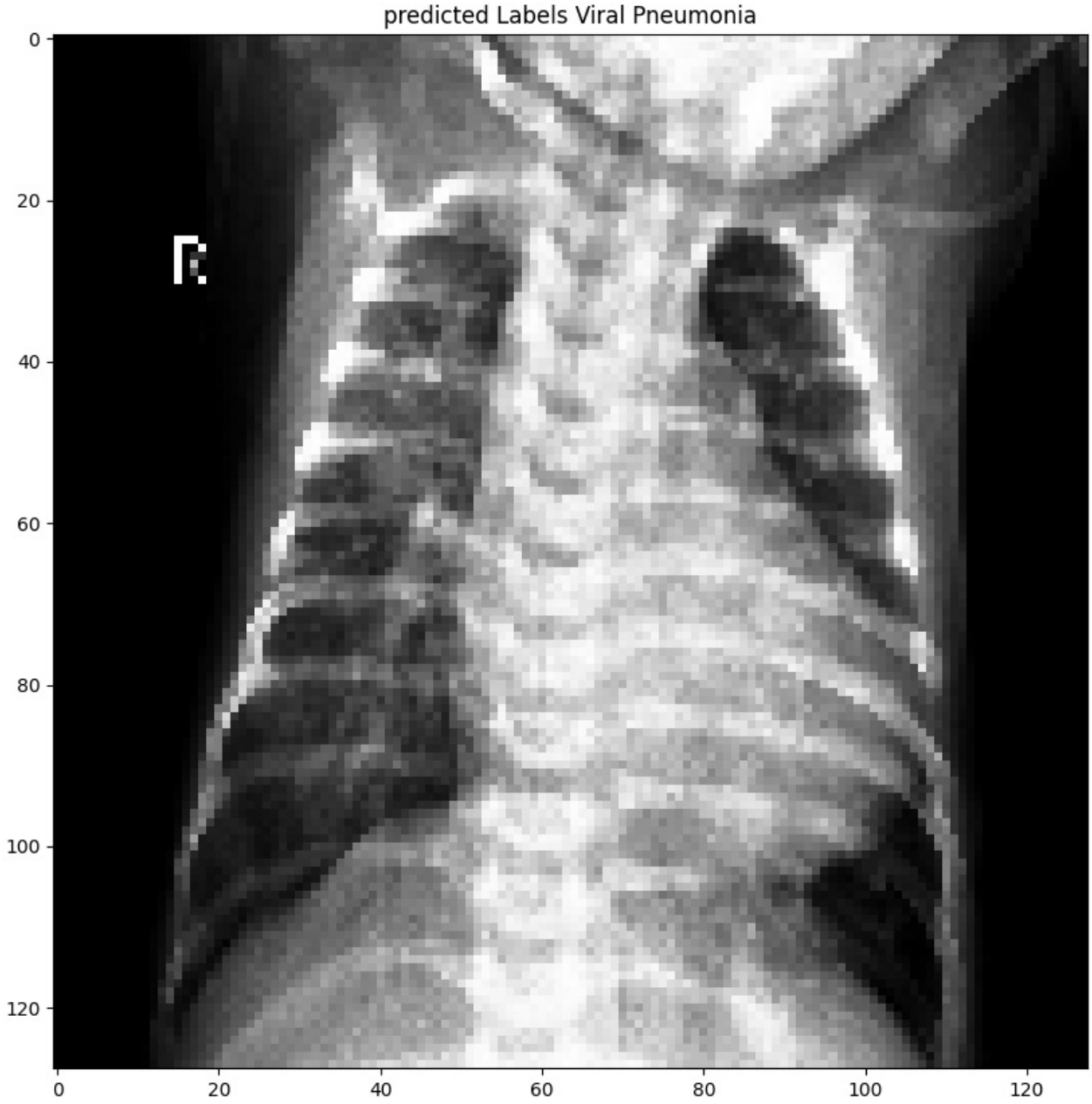

predicted Labels Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120
F

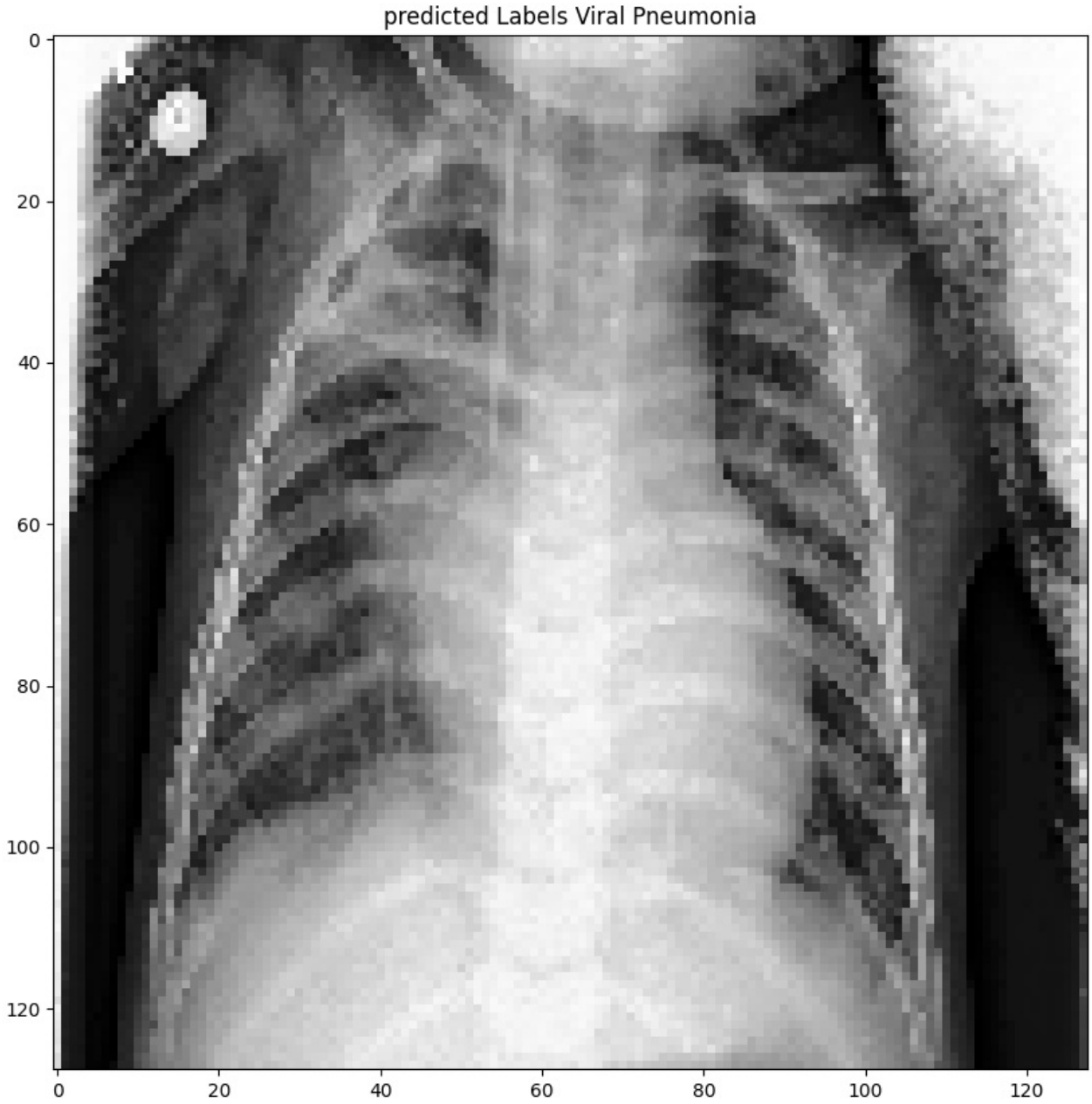

predicted Labels Viral Pneumonia
0
20
40
60
80
100
120
0
20
40
60
80
100
120

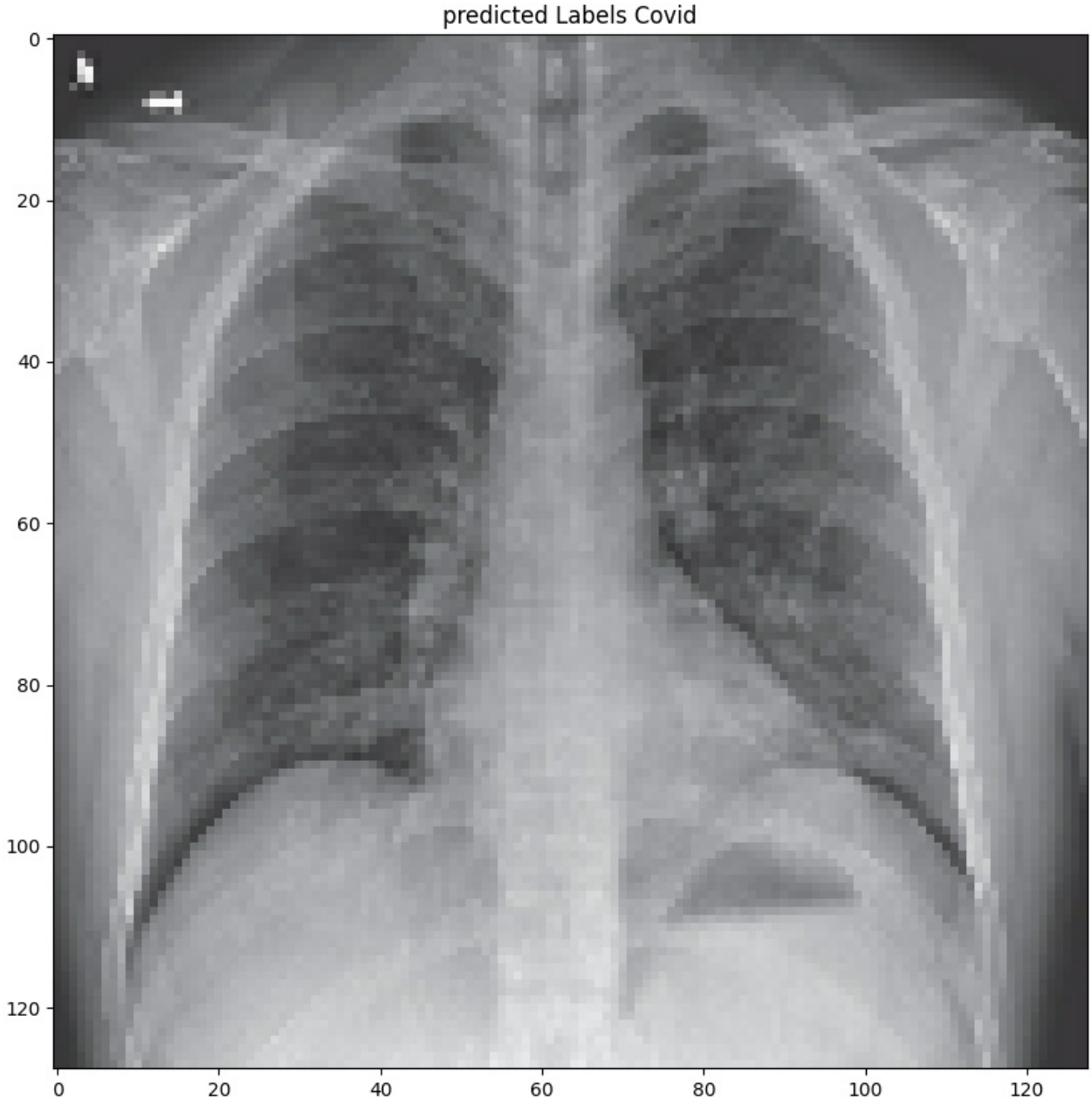

predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

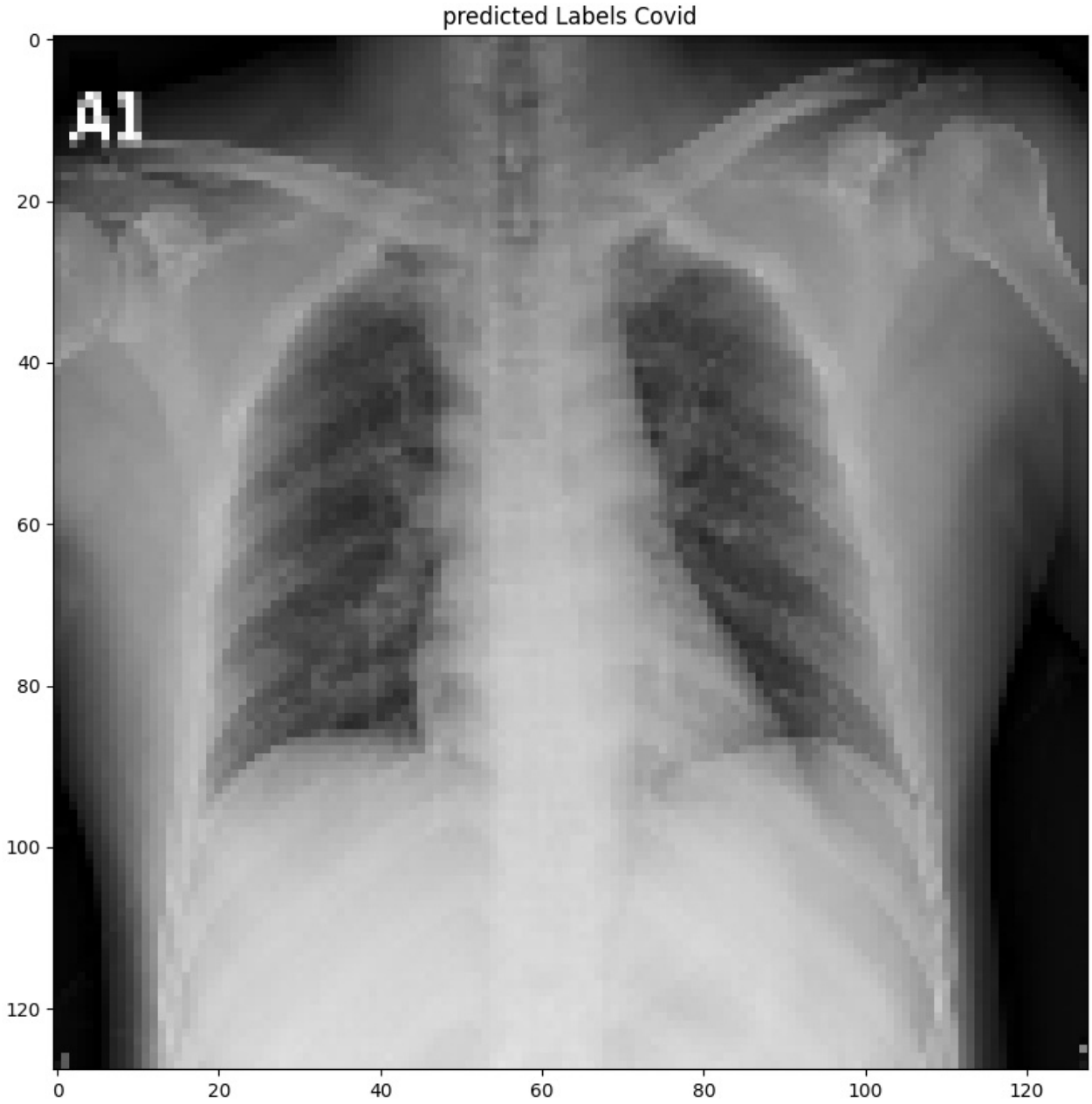

predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

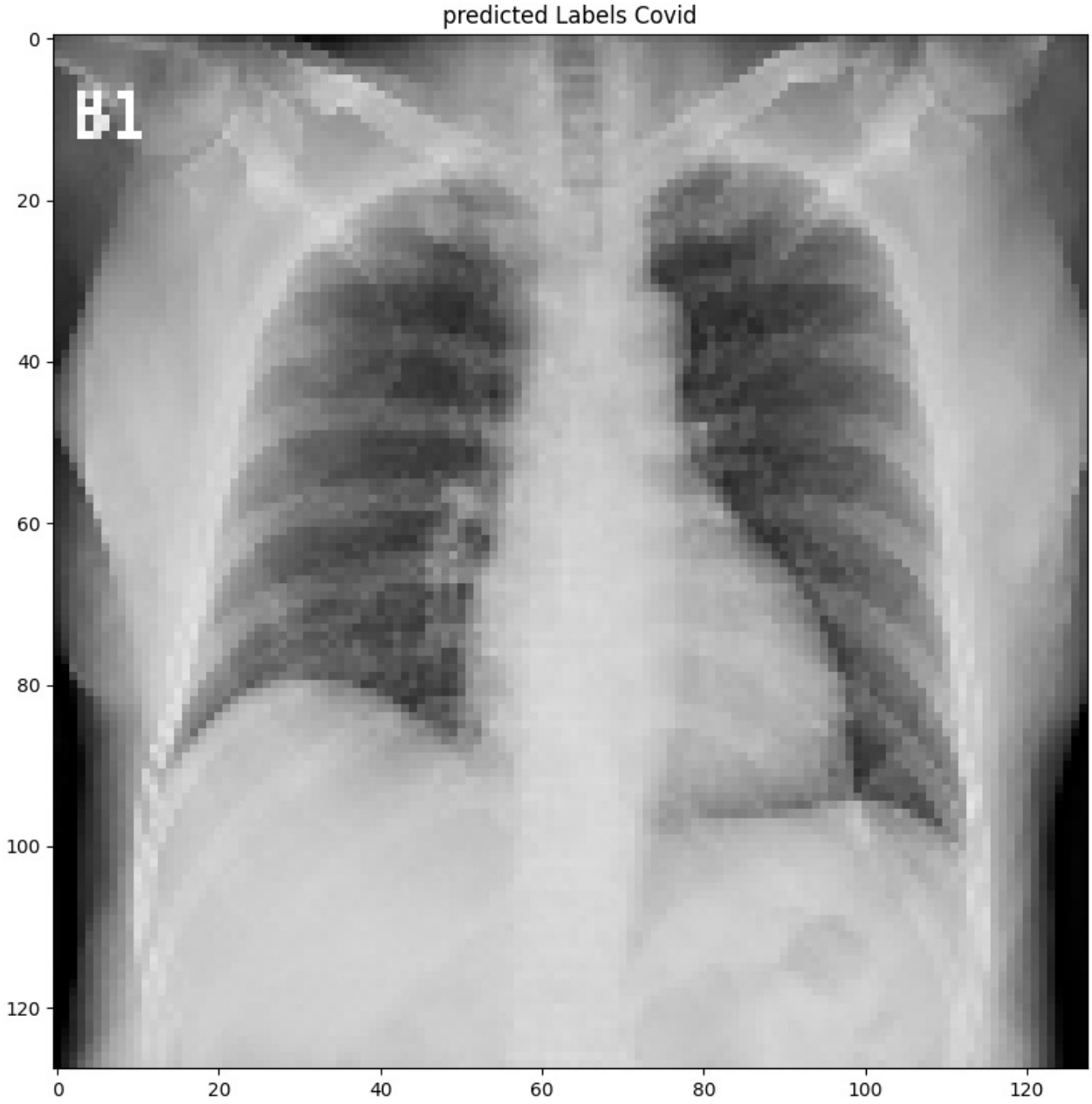

predicted Labels Viral Pneumonia
0
20
40
60
80
100
120
0
20
40
60
80
100
120

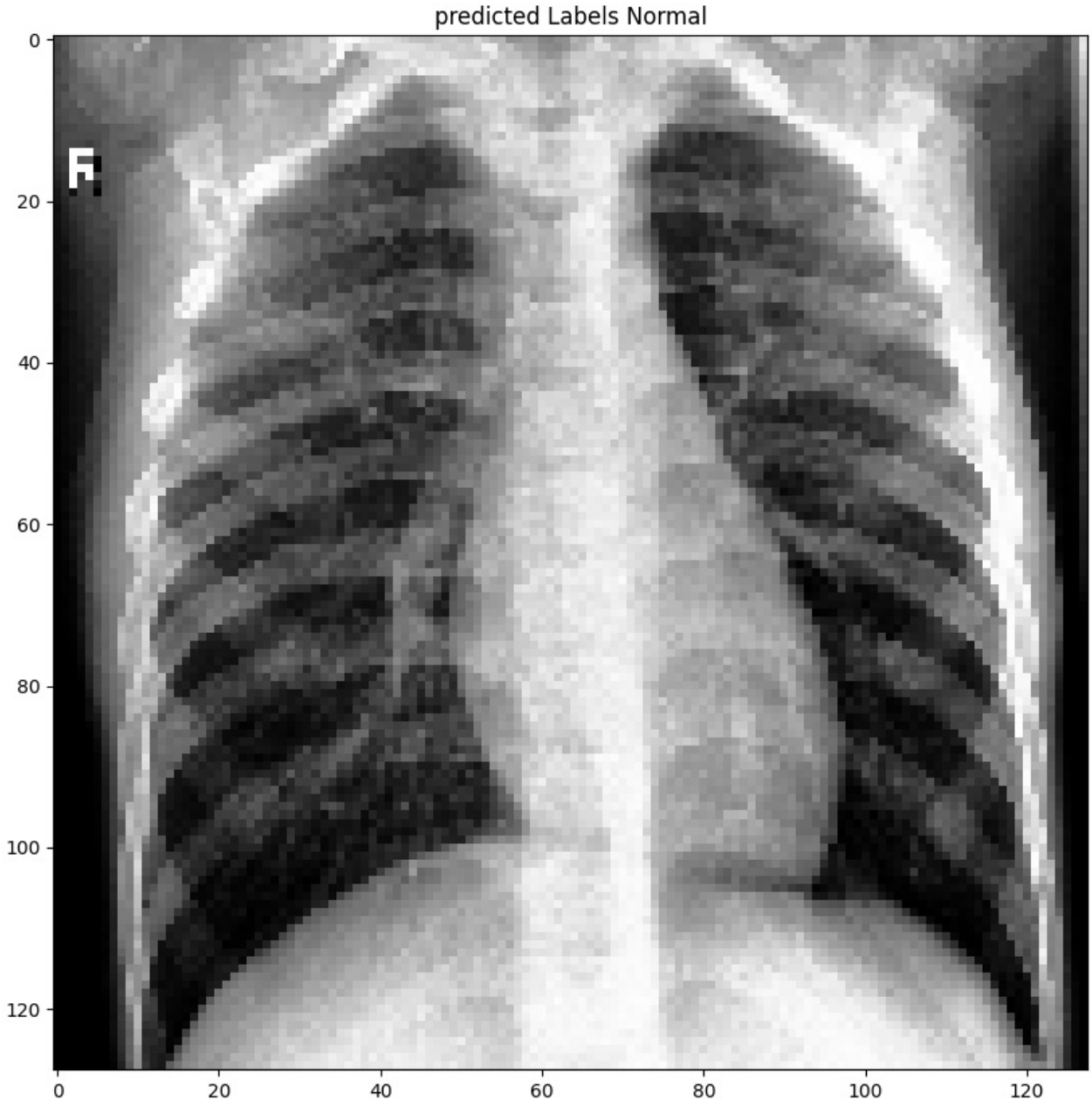

predicted Labels Viral Pneumonia
0
20
40
60
80
100
120
0
20
40
60
80
100
120

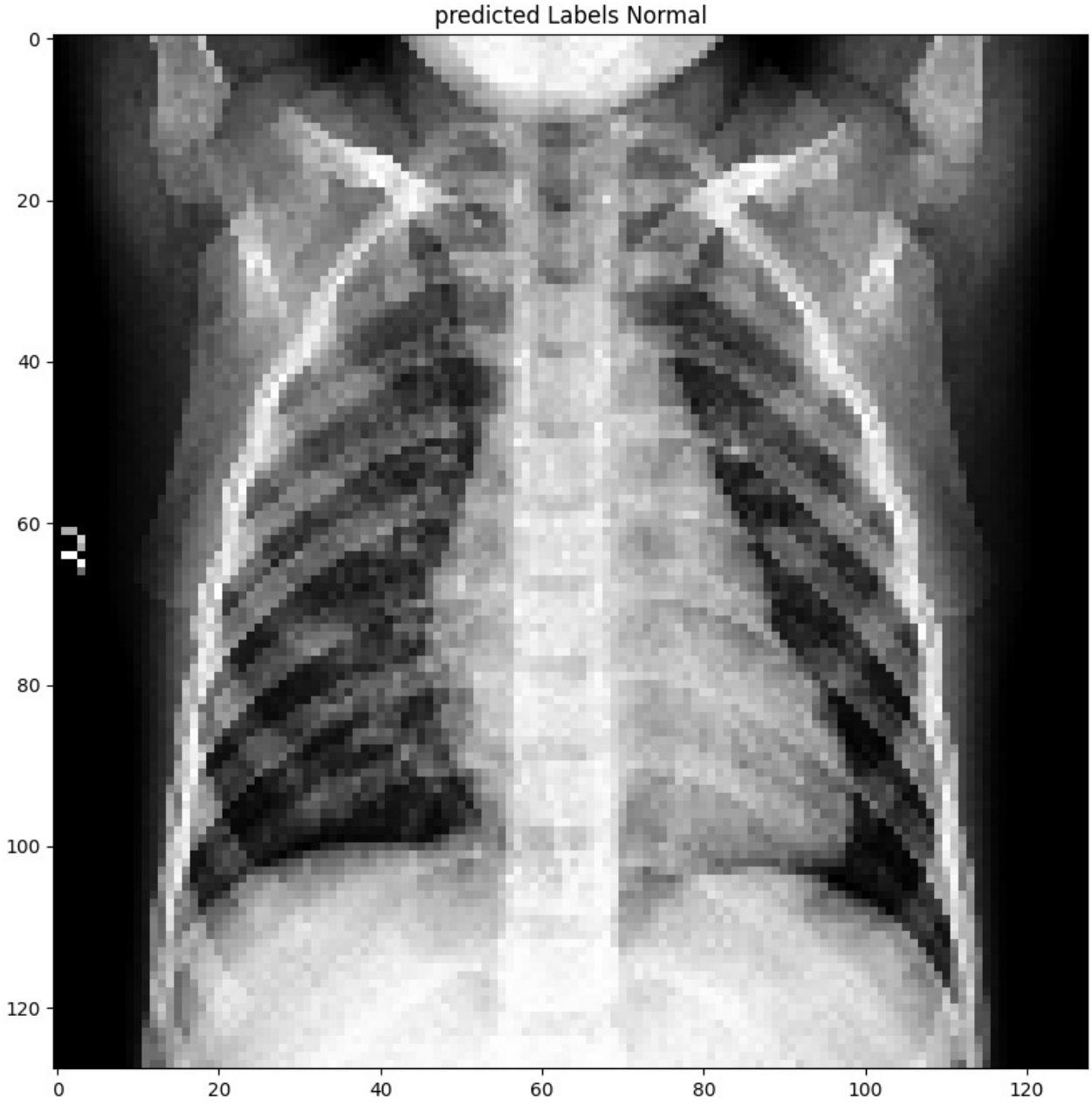

predicted Labels Viral Pneumonia
0
20
40
60
80
100
120
0
20
40
60
80
100
120

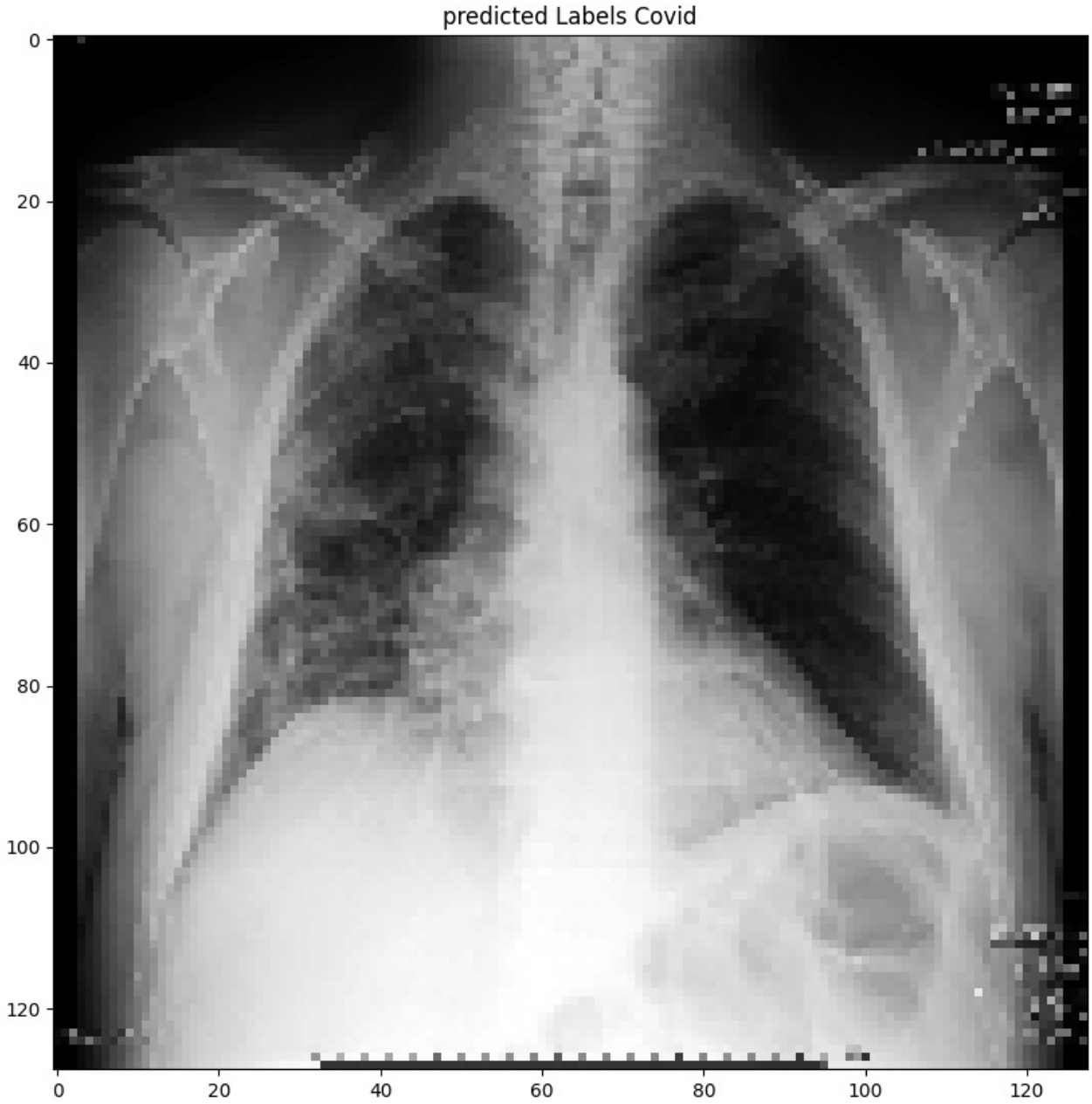

predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

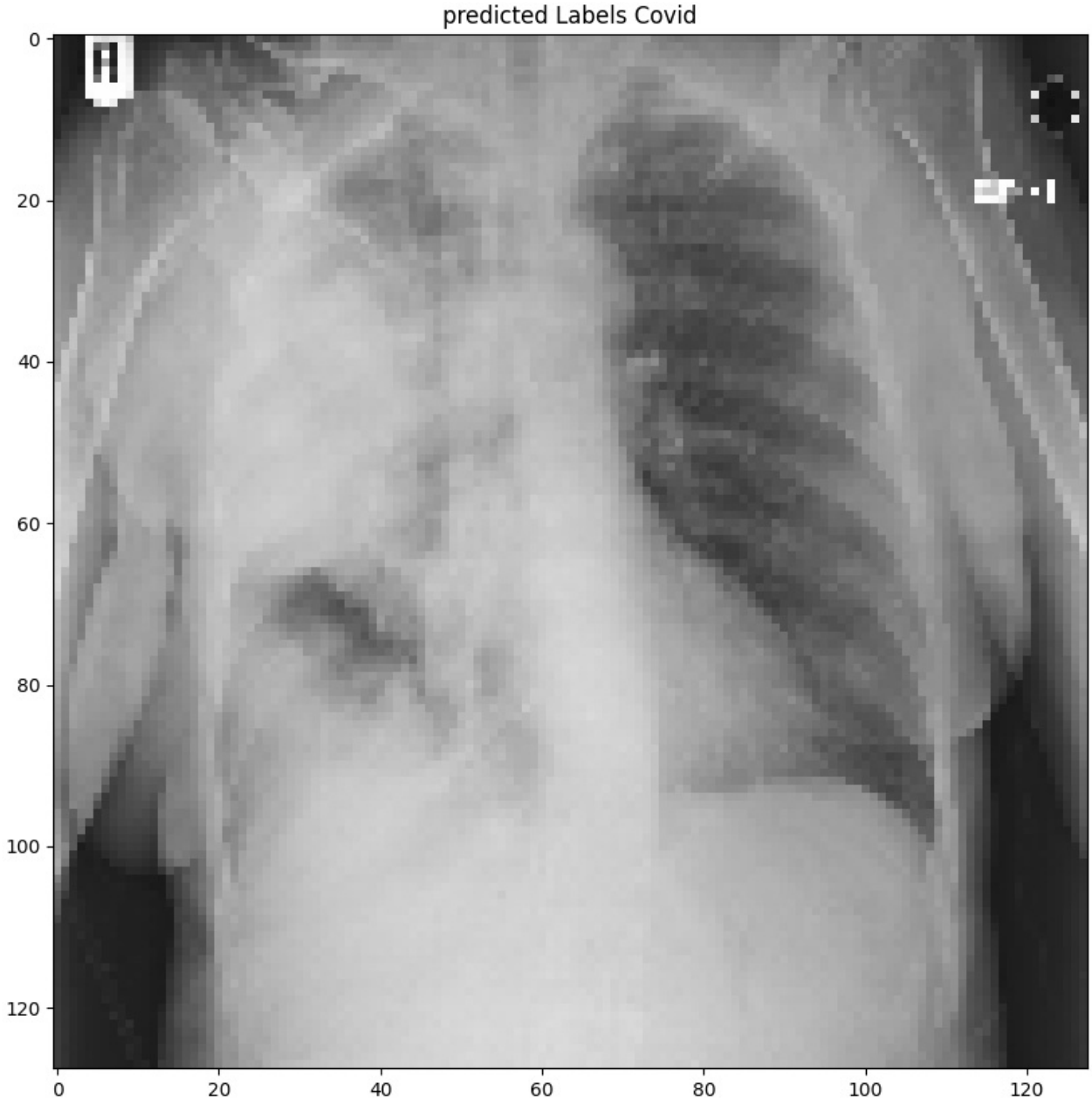

predicted Labels Covid
A1
0
20
40
60
80
100
120
0
20
40
60
80
100
120

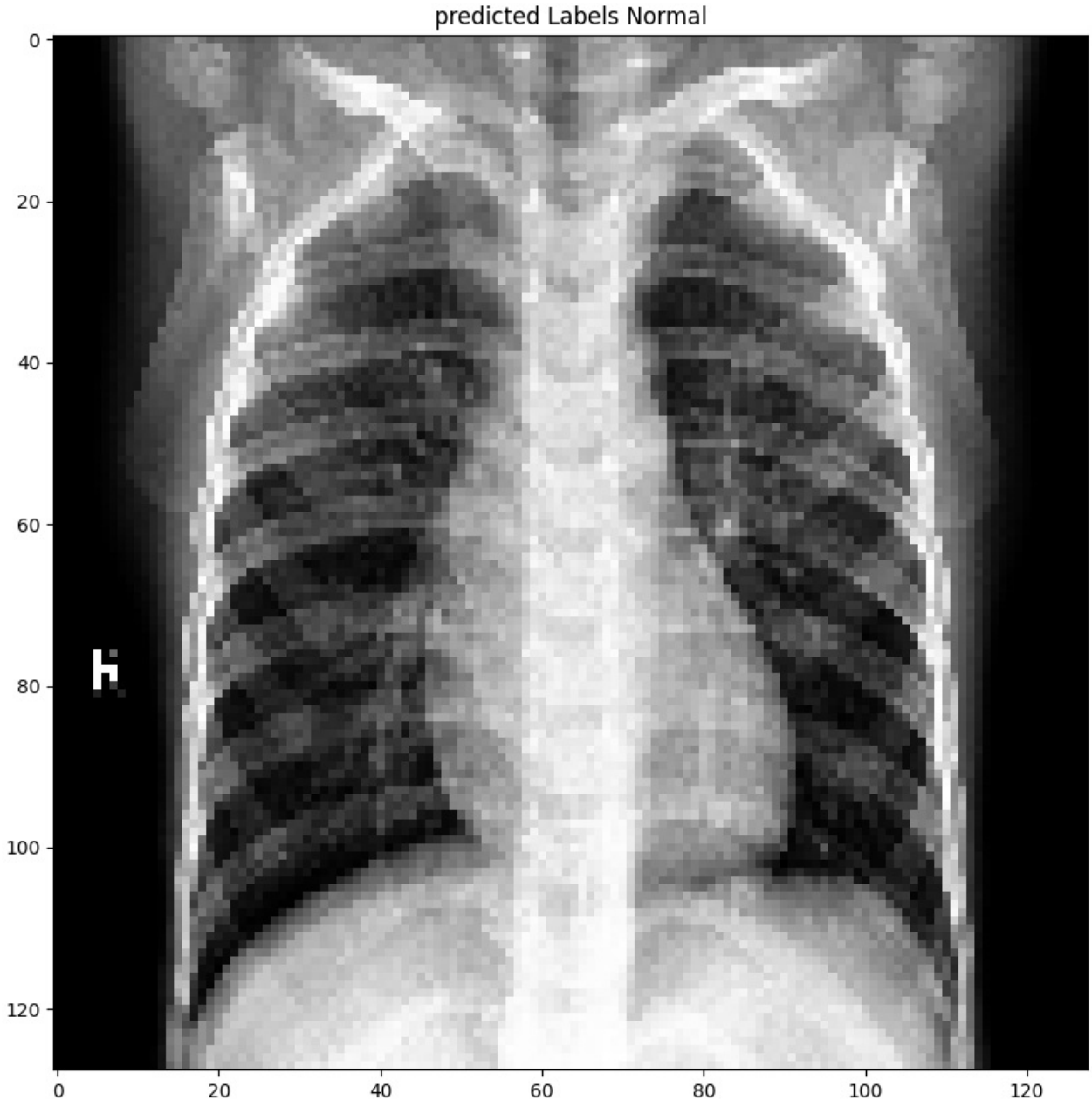

predicted Labels Covid
B1
0
20
40
60
80
100
120
0
20
40
60
80
100
120

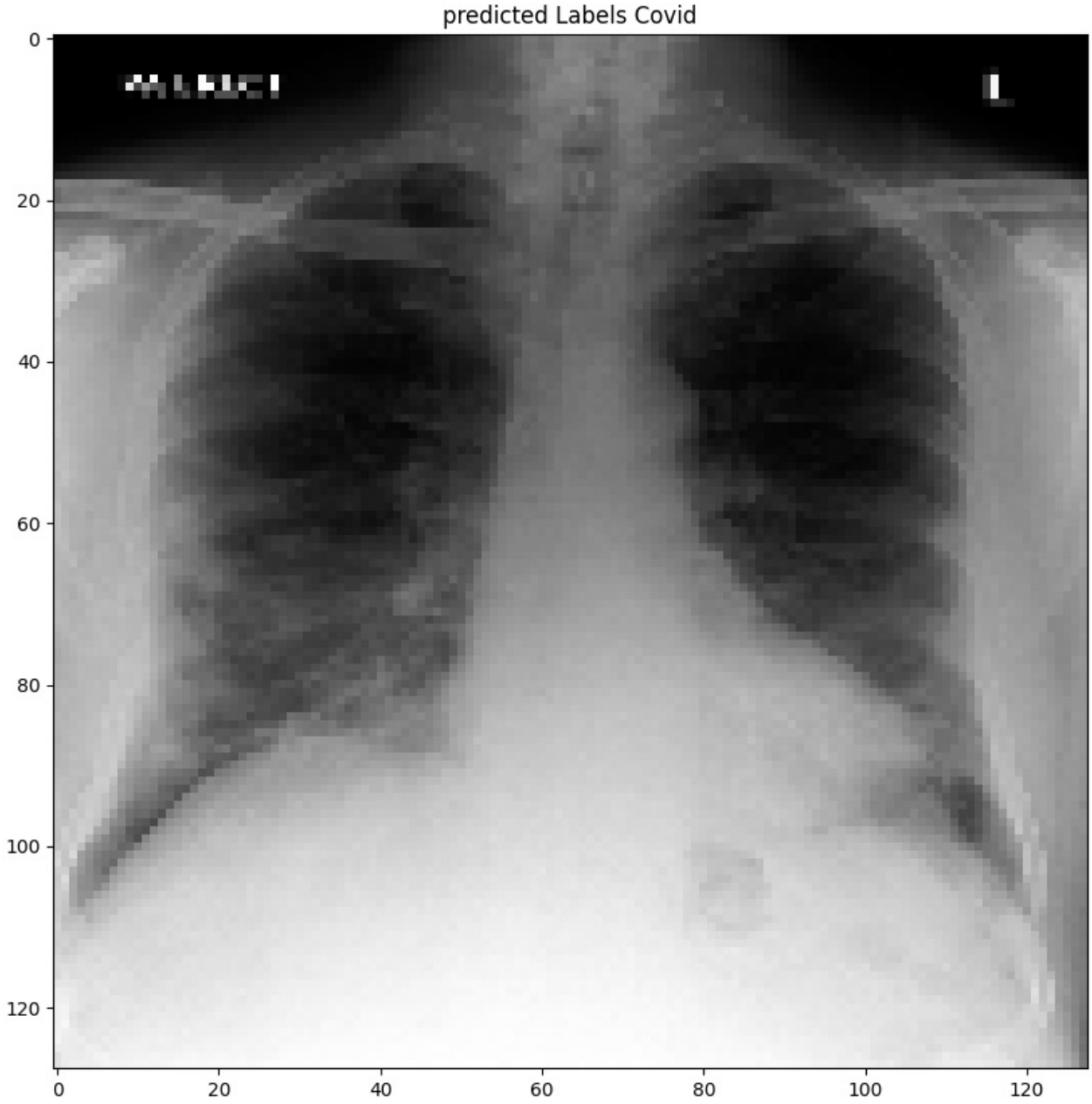

predicted Labels Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

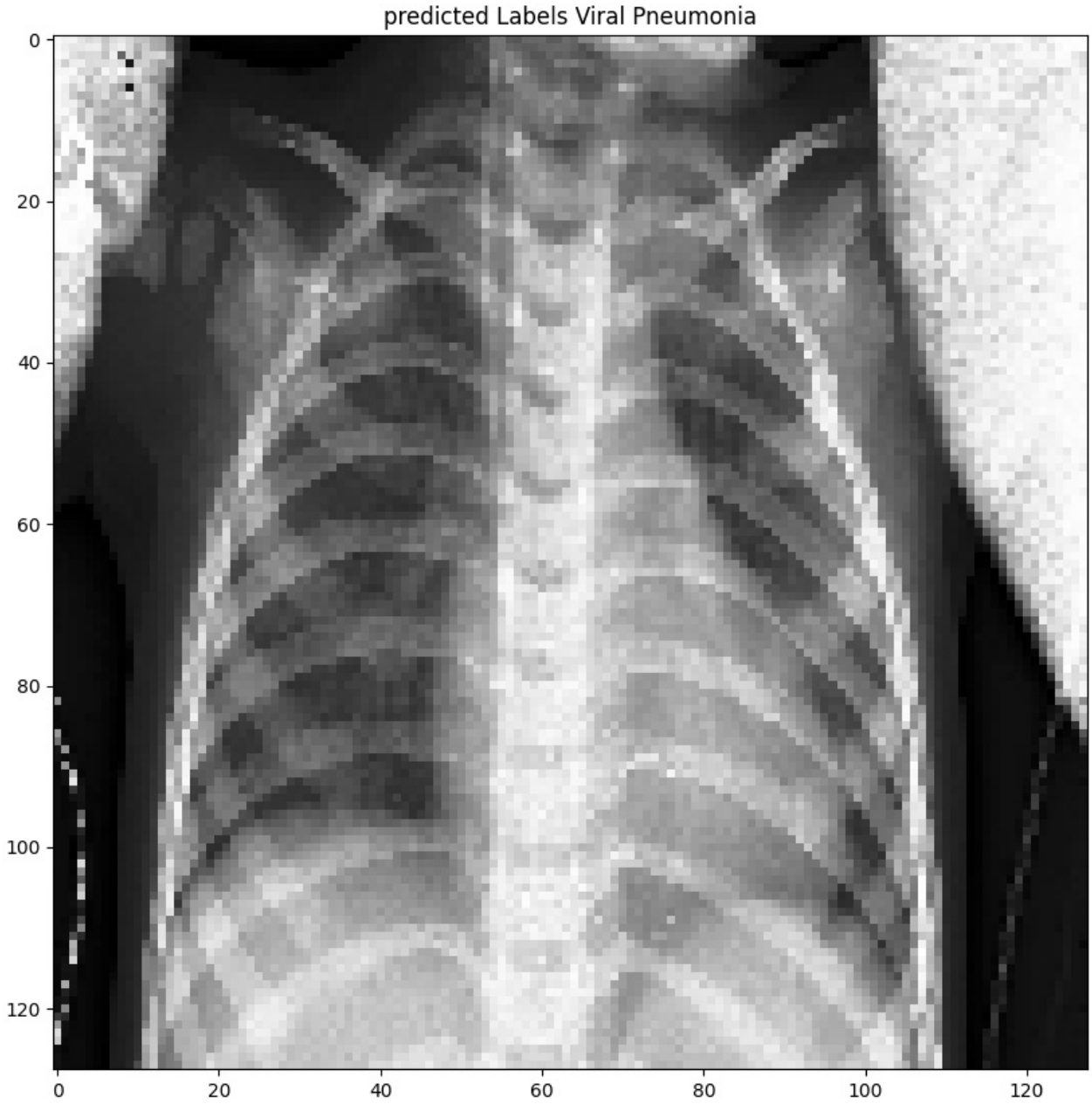

predicted Labels Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

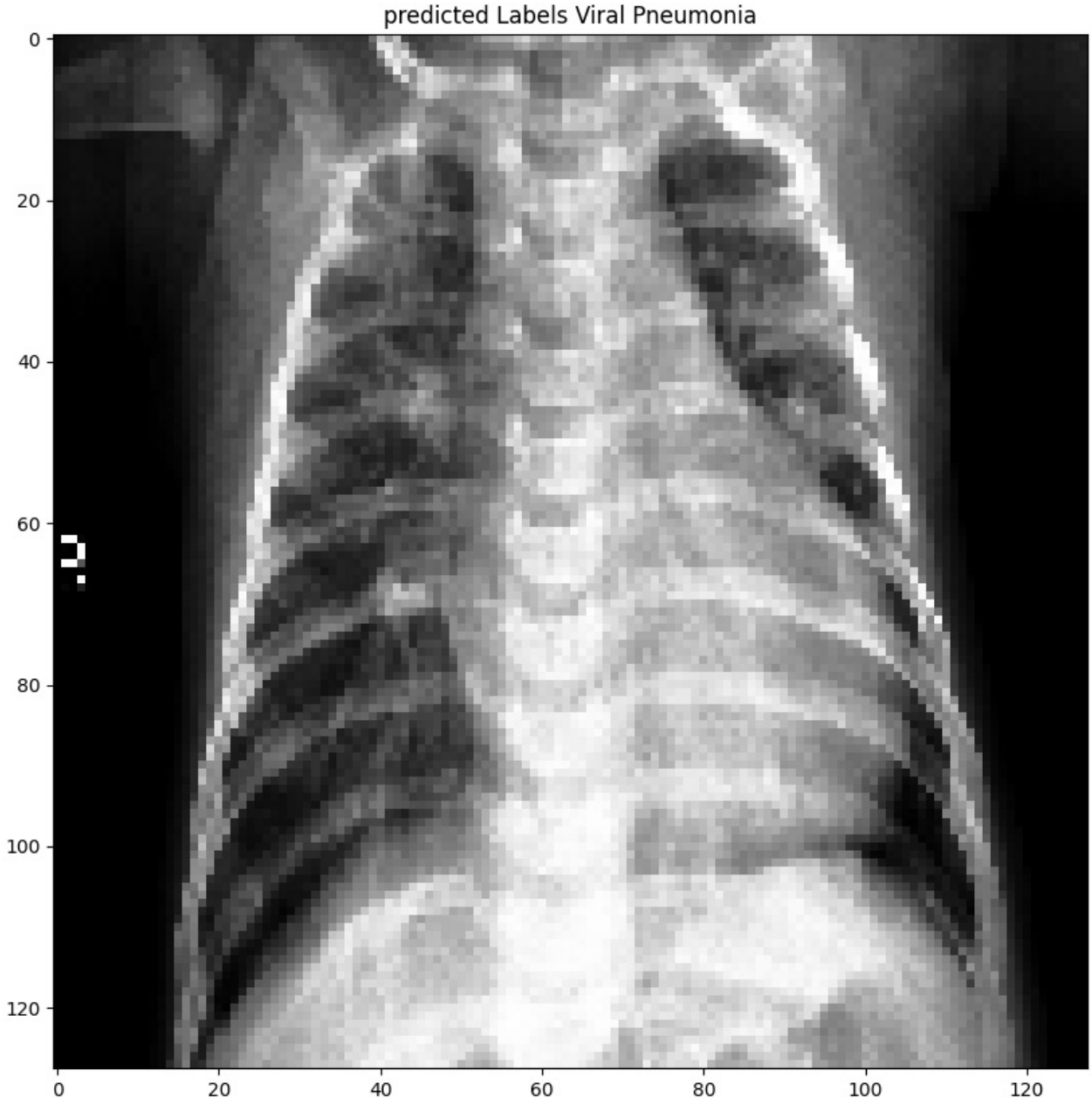

predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

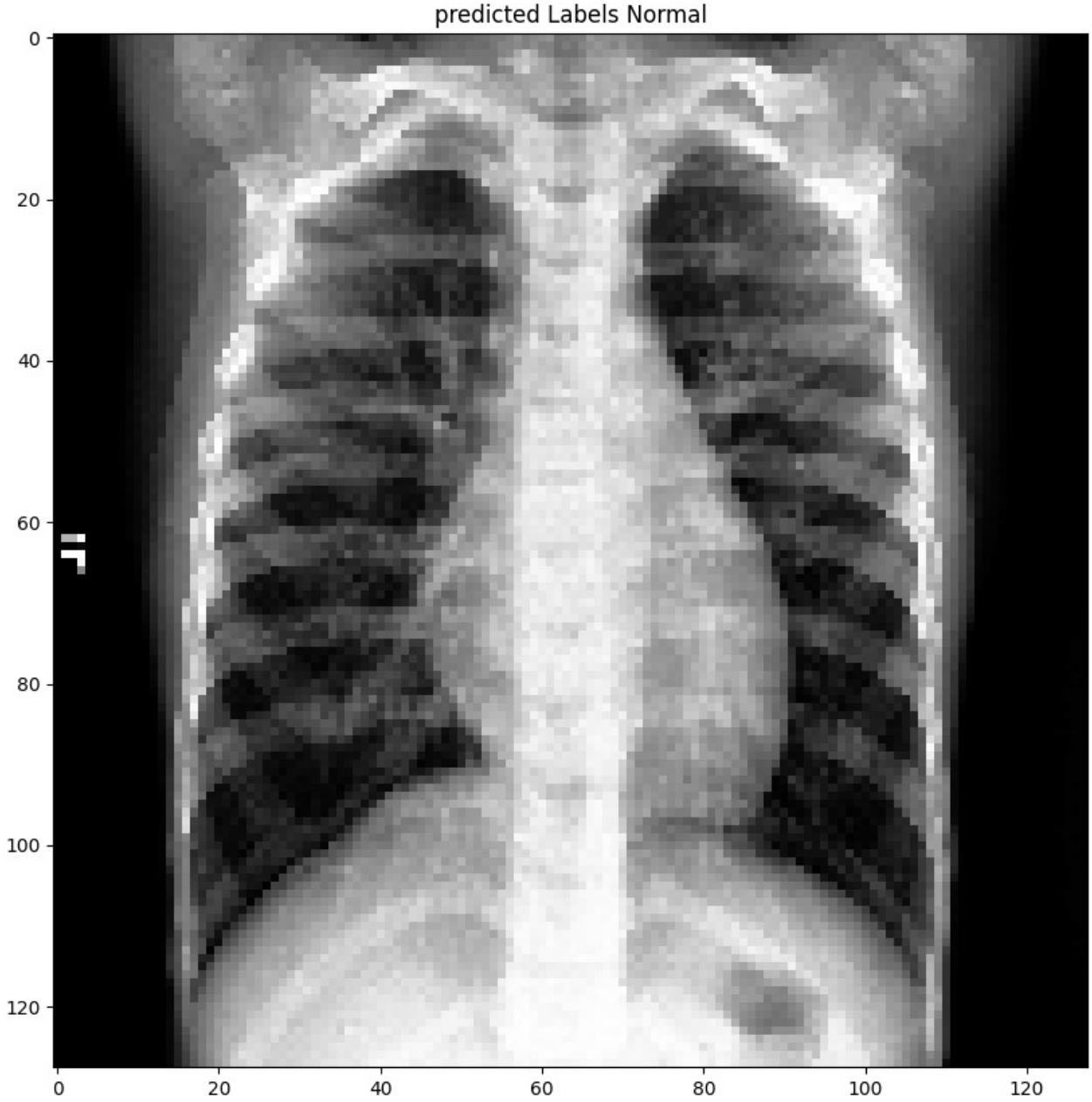

predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120

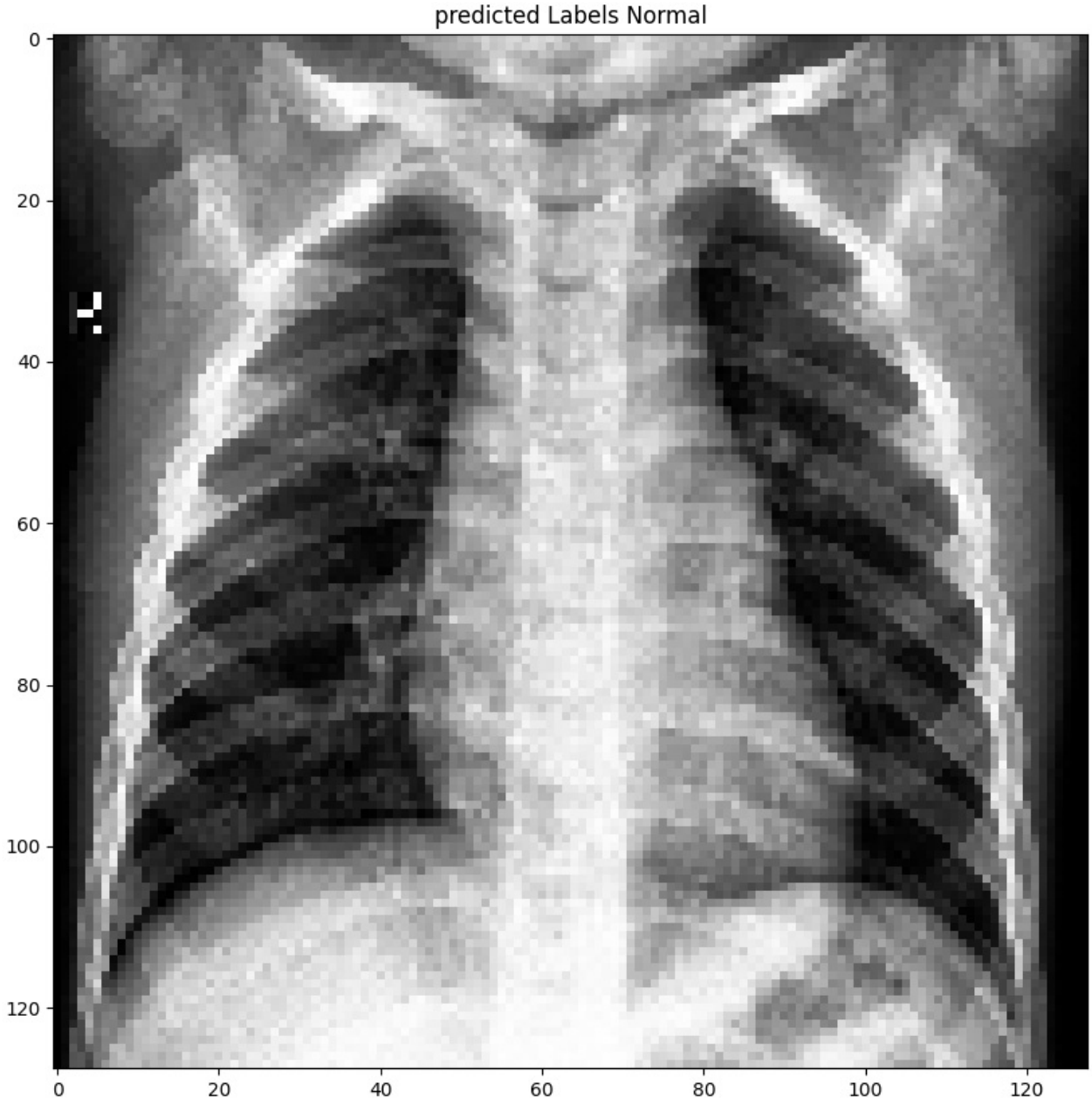

predicted Labels Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120


predicted Labels Covid
0
20
40
60
80
100
120
0
20
40
60
80
100
120


predicted Labels Viral Pneumonia
0
20
40
60
80
100
120
0
20
40
60
80
100
120


predicted Labels Viral Pneumonia
0
20
40
60
80
100
120
0
20
40
60
80
100
120


predicted Labels Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120


predicted Labels Normal
0
20
40
60
80
100
120
0
20
40
60
80
100
120

In [ ]:

Loading [MathJax]/extensions/Safe.js