In [1]:

```
!pip install gradio
```

```
Requirement already satisfied: gradio in c:\users\paturu vineetha\anaconda3\lib\site-packages (3.15.0)
Requirement already satisfied: fastapi in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (0.88.0)
Requirement already satisfied: uvicorn in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (0.20.0)
Requirement already satisfied: aiohttp in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (3.8.3)
Requirement already satisfied: markdown-it-py[linkify,plugins] in c:\users\paturu vineetha\anaconda3\lib\site-packages (fro
m gradio) (2.1.0)
Requirement already satisfied: jinja2 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (2.11.3)
Requirement already satisfied: orjson in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (3.8.3)
Requirement already satisfied: pydantic in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (1.10.4)
Requirement already satisfied: pyyaml in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (6.0)
Requirement already satisfied: altair>=4.2.0 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (4.2.0)
Requirement already satisfied: requests in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (2.26.0)
Requirement already satisfied: pillow in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (8.4.0)
Requirement already satisfied: pandas in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (1.3.4)
Requirement already satisfied: pydub in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (0.25.1)
Requirement already satisfied: ffmpy in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (0.3.0)
Requirement already satisfied: numpy in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (1.20.3)
Requirement already satisfied: markupsafe in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (1.1.1)
Requirement already satisfied: python-multipart in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (0.0.
5)
Requirement already satisfied: matplotlib in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (3.4.3)
Requirement already satisfied: fsspec in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (2021.10.1)
Requirement already satisfied: websockets>=10.0 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (10.
4)
Requirement already satisfied: pycryptodome in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (3.16.0)
Requirement already satisfied: httpx in c:\users\paturu vineetha\anaconda3\lib\site-packages (from gradio) (0.23.1)
Requirement already satisfied: jsonschema>=3.0 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from altair>=4.2.0-
>gradio) (3.2.0)
Requirement already satisfied: toolz in c:\users\paturu vineetha\anaconda3\lib\site-packages (from altair>=4.2.0->gradio)
(0.11.1)
Requirement already satisfied: entrypoints in c:\users\paturu vineetha\anaconda3\lib\site-packages (from altair>=4.2.0->gra
dio) (0.3)
Requirement already satisfied: six>=1.11.0 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from jsonschema>=3.0->a
ltair>=4.2.0->gradio) (1.16.0)
Requirement already satisfied: setuptools in c:\users\paturu vineetha\anaconda3\lib\site-packages (from jsonschema>=3.0->al
tair>=4.2.0->gradio) (58.0.4)
Requirement already satisfied: pyrsistent>=0.14.0 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from jsonschema>
=3.0->altair>=4.2.0->gradio) (0.18.0)
Requirement already satisfied: attrs>=17.4.0 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from jsonschema>=3.0-
>altair>=4.2.0->gradio) (21.2.0)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from pandas-
>gradio) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from pandas->gradio)
(2021.3)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from ai
ohttp->gradio) (4.0.2)
Requirement already satisfied: charset-normalizer<3.0,>=2.0 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from a
iohttp->gradio) (2.0.4)
Requirement already satisfied: yarl<2.0,>=1.0 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from aiohttp->gradi
o) (1.8.2)
Requirement already satisfied: multidict<7.0,>=4.5 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from aiohttp->g
radio) (6.0.4)
Requirement already satisfied: frozenlist>=1.1.1 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from aiohttp->gra
dio) (1.3.3)
Requirement already satisfied: aiosignal>=1.1.2 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from aiohttp->grad
io) (1.3.1)
Requirement already satisfied: idna>=2.0 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from yarl<2.0,>=1.0->aioh
ttp->gradio) (3.2)
Requirement already satisfied: starlette==0.22.0 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from fastapi->gra
dio) (0.22.0)
Requirement already satisfied: anyio<5,>=3.4.0 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from starlette==0.2
2.0->fastapi->gradio) (3.6.2)
Requirement already satisfied: typing-extensions>=3.10.0 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from star
lette==0.22.0->fastapi->gradio) (4.4.0)
Requirement already satisfied: sniffio>=1.1 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from anyio<5,>=3.4.0->
starlette==0.22.0->fastapi->gradio) (1.2.0)
Requirement already satisfied: rfc3986[idna2008]<2,>=1.3 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from http
x->gradio) (1.5.0)
Requirement already satisfied: httpcore<0.17.0,>=0.15.0 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from httpx
->gradio) (0.16.3)
Requirement already satisfied: certifi in c:\users\paturu vineetha\anaconda3\lib\site-packages (from httpx->gradio) (2021.1
0.8)
Requirement already satisfied: h11<0.15,>=0.13 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from httpcore<0.17.
0,>=0.15.0->httpx->gradio) (0.14.0)
Requirement already satisfied: mdurl~=0.1 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from markdown-it-py[link
ify,plugins]->gradio) (0.1.2)
Requirement already satisfied: linkify-it-py~=1.0 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from markdown-it
-py[linkify,plugins]->gradio) (1.0.3)
Requirement already satisfied: mdit-py-plugins in c:\users\paturu vineetha\anaconda3\lib\site-packages (from markdown-it-py
[linkify,plugins]->gradio) (0.3.3)
Requirement already satisfied: uc-micro-py in c:\users\paturu vineetha\anaconda3\lib\site-packages (from linkify-it-py~=1.0
->markdown-it-py[linkify,plugins]->gradio) (1.0.1)
Requirement already satisfied: cycler>=0.10 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from matplotlib->gradi
o) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from matplotlib->
gradio) (1.3.1)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from matplotlib->g
radio) (3.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from requests
->gradio) (1.26.7)
Requirement already satisfied: click>=7.0 in c:\users\paturu vineetha\anaconda3\lib\site-packages (from uvicorn->gradio)
```

```
(8.0.3)
Requirement already satisfied: colorama in c:\users\paturu vineetha\anaconda3\lib\site-packages (from click>=7.0->uvicorn->
gradio) (0.4.4)
```

In [2]:

```
import matplotlib.pyplot as plt
import seaborn as sns
import keras
from keras.models import Sequential
from keras.layers import Dense, Conv2D , MaxPool2D , Flatten , Dropout , BatchNormalization
from keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report,confusion_matrix
from keras.callbacks import ReduceLROnPlateau
import cv2
import os
import numpy as np
import pandas as pd
import gradio
```

In [3]:

```
labels = ['ARDS', 'NORMAL']
img_size = 150
def get_data(data_dir):
    data = []
    for label in labels:
        path = os.path.join(data_dir, label)
        class_num = labels.index(label)
        for img in os.listdir(path):
            try:
                img_arr = cv2.imread(os.path.join(path, img), cv2.IMREAD_GRAYSCALE)
                resized_arr = cv2.resize(img_arr, (img_size, img_size)) # Reshaping images to preferred size
                data.append([resized_arr, class_num])
            except Exception as e:
                print(e)
    return np.array(data)
```

In [5]:

```
train = get_data(r'C:\Users\paturu vineetha\Dropbox\PC\Desktop\train')
test = get_data(r'C:\Users\paturu vineetha\Dropbox\PC\Desktop\test')
val = get_data(r'C:\Users\paturu vineetha\Dropbox\PC\Desktop\val')
```

```
C:\Users\PATURU~1\AppData\Local\Temp/ipykernel_5436/3622272874.py:15: VisibleDeprecationWarning: Creating an ndarray from r
agged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprec
ated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.
  return np.array(data)
```

In [6]:

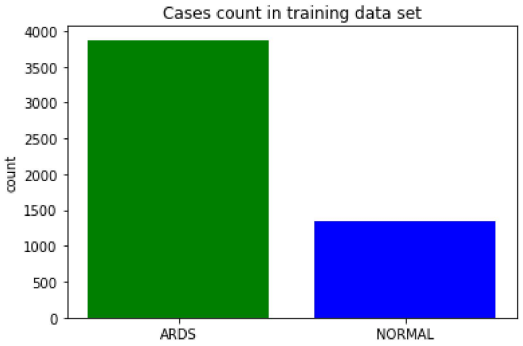
```
x_train = []
y_train = []

x_val = []
y_val = []

x_test = []
y_test = []

for feature, label in train:
    x_train.append(feature)
    y_train.append(label)

for feature, label in test:
    x_test.append(feature)
    y_test.append(label)

for feature, label in val:
    x_val.append(feature)
    y_val.append(label)
```

In [7]:

```
positives=[]
negatives=[]
for i in range(len(y_train)):
    if y_train[i]:
        positives.append(x_train[i])
    else:
        negatives.append(x_train[i])
```

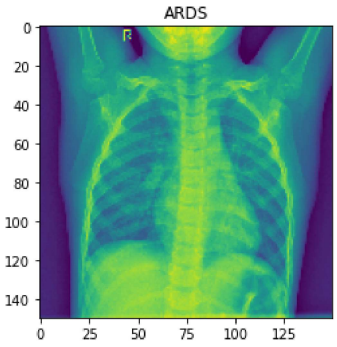
In [8]:

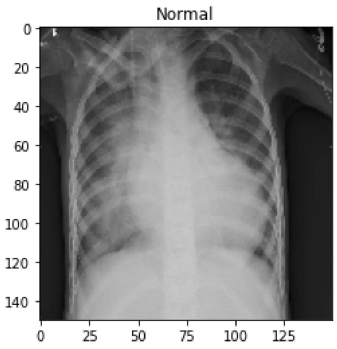
```
plt.bar(labels, [len(negatives), len(positives)], color=["green", "blue"])
plt.title("Cases count in training data set")
plt.ylabel("count")
plt.show()
```



In [9]:

```
plt.imshow(positives[0])
plt.title("ARDS")
plt.show()

plt.imshow(negatives[4], cmap="gray")
plt.title("Normal")
plt.show()
```





In [10]:

```
# Normalize the data
x_train = np.array(x_train) / 255
x_val = np.array(x_val) / 255
x_test = np.array(x_test) / 255
```

In [11]:

```
# resize data for deep learning
x_train = x_train.reshape(-1, img_size, img_size, 1)
y_train = np.array(y_train)

x_val = x_val.reshape(-1, img_size, img_size, 1)
y_val = np.array(y_val)

x_test = x_test.reshape(-1, img_size, img_size, 1)
y_test = np.array(y_test)
```

In [12]:

```
x_test[0].shape
```

Out[12]:

```
(150, 150, 1)
```

In [13]:

```
y_train = y_train.reshape(-1,1)
y_test = y_test.reshape(-1,1)
y_val = y_val.reshape(-1,1)
```

In [14]:

```
datagen = ImageDataGenerator(
        featurewise_center=False,
        samplewise_center=False,
        featurewise_std_normalization=False,
        samplewise_std_normalization=False,
        zca_whitening=False,
        rotation_range = 30,  # randomly rotate images in the range (degrees, 0 to 180)
        zoom_range = 0.2, # Randomly zoom image
        width_shift_range=0.1,  # randomly shift images horizontally (fraction of total width)
        height_shift_range=0.1,  # randomly shift images vertically (fraction of total height)
        horizontal_flip = True,  # randomly flip images
        vertical_flip=False)

datagen.fit(x_train)
```

In [15]:

```
model = Sequential()
model.add(Conv2D(32 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu' , input_shape = (150,150,1)))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))

model.add(Conv2D(64 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(Dropout(0.1))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))

model.add(Conv2D(64 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))

model.add(Conv2D(128 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))

model.add(Conv2D(256 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))

model.add(Flatten())
model.add(Dense(units = 128 , activation = 'relu'))
model.add(Dropout(0.2))
model.add(Dense(units = 1 , activation = 'sigmoid'))
model.compile(optimizer = "rmsprop" , loss = 'binary_crossentropy' , metrics = ['accuracy'])
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 150, 150, 32)      320

 batch_normalization (BatchN  (None, 150, 150, 32)     128
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 75, 75, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 75, 75, 64)        18496

 dropout (Dropout)           (None, 75, 75, 64)        0

 batch_normalization_1 (Batc  (None, 75, 75, 64)       256
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 38, 38, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 38, 38, 64)        36928

 batch_normalization_2 (Batc  (None, 38, 38, 64)       256
 hNormalization)

 max_pooling2d_2 (MaxPooling  (None, 19, 19, 64)       0
 2D)

 conv2d_3 (Conv2D)           (None, 19, 19, 128)       73856

 dropout_1 (Dropout)         (None, 19, 19, 128)       0

 batch_normalization_3 (Batc  (None, 19, 19, 128)      512
 hNormalization)

 max_pooling2d_3 (MaxPooling  (None, 10, 10, 128)      0
 2D)

 conv2d_4 (Conv2D)           (None, 10, 10, 256)       295168

 dropout_2 (Dropout)         (None, 10, 10, 256)       0

 batch_normalization_4 (Batc  (None, 10, 10, 256)      1024
 hNormalization)

 max_pooling2d_4 (MaxPooling  (None, 5, 5, 256)        0
 2D)

 flatten (Flatten)           (None, 6400)              0

 dense (Dense)               (None, 128)               819328

 dropout_3 (Dropout)         (None, 128)               0

 dense_1 (Dense)             (None, 1)                 129

=================================================================
Total params: 1,246,401
Trainable params: 1,245,313
Non-trainable params: 1,088
_________________________________________________________________
```

In [16]:

```
model.compile(optimizer = "rmsprop" ,
            loss = 'binary_crossentropy' ,
            metrics = ['accuracy'])
# model.summary()
```

In [17]:

```
learning_rate_reduction = ReduceLROnPlateau(monitor='val_accuracy',
                                            patience = 2,
                                            verbose=1,
                                            factor=0.3,
                                            min_lr=0.000001)
```

In [18]:

```
history = model.fit(datagen.flow(x_train,y_train, batch_size = 32) ,
                    epochs = 10 ,
                    validation_data = datagen.flow(x_val, y_val) ,
                    callbacks = learning_rate_reduction)
```

```
Epoch 1/10
163/163 [==============================] - 259s 2s/step - loss: 0.4903 - accuracy: 0.8441 - val_loss: 24.4454 - val_accurac
y: 0.5000 - lr: 0.0010
Epoch 2/10
163/163 [==============================] - 249s 2s/step - loss: 0.2723 - accuracy: 0.8992 - val_loss: 46.2482 - val_accurac
y: 0.5000 - lr: 0.0010
Epoch 3/10
163/163 [==============================] - ETA: 0s - loss: 0.2095 - accuracy: 0.9214
Epoch 3: ReduceLROnPlateau reducing learning rate to 0.0003000000142492354.
163/163 [==============================] - 234s 1s/step - loss: 0.2095 - accuracy: 0.9214 - val_loss: 18.2159 - val_accurac
y: 0.5000 - lr: 0.0010
Epoch 4/10
163/163 [==============================] - 228s 1s/step - loss: 0.1561 - accuracy: 0.9444 - val_loss: 42.8608 - val_accurac
y: 0.5000 - lr: 3.0000e-04
Epoch 5/10
163/163 [==============================] - 237s 1s/step - loss: 0.1348 - accuracy: 0.9534 - val_loss: 1.9288 - val_accurac
y: 0.5625 - lr: 3.0000e-04
Epoch 6/10
163/163 [==============================] - 240s 1s/step - loss: 0.1181 - accuracy: 0.9571 - val_loss: 7.6320 - val_accurac
y: 0.5000 - lr: 3.0000e-04
Epoch 7/10
163/163 [==============================] - 226s 1s/step - loss: 0.1298 - accuracy: 0.9548 - val_loss: 2.2881 - val_accurac
y: 0.6250 - lr: 3.0000e-04
Epoch 8/10
163/163 [==============================] - 235s 1s/step - loss: 0.1291 - accuracy: 0.9590 - val_loss: 0.6501 - val_accurac
y: 0.7500 - lr: 3.0000e-04
Epoch 9/10
163/163 [==============================] - 241s 1s/step - loss: 0.1207 - accuracy: 0.9584 - val_loss: 0.4952 - val_accurac
y: 0.6875 - lr: 3.0000e-04
Epoch 10/10
163/163 [==============================] - ETA: 0s - loss: 0.1064 - accuracy: 0.9632
Epoch 10: ReduceLROnPlateau reducing learning rate to 9.000000427477062e-05.
163/163 [==============================] - 285s 2s/step - loss: 0.1064 - accuracy: 0.9632 - val_loss: 28.0817 - val_accurac
y: 0.5000 - lr: 3.0000e-04
```

In [19]:

```
model.save_weights('kaggle/saved_model_ai/pneumoniadetection')
```

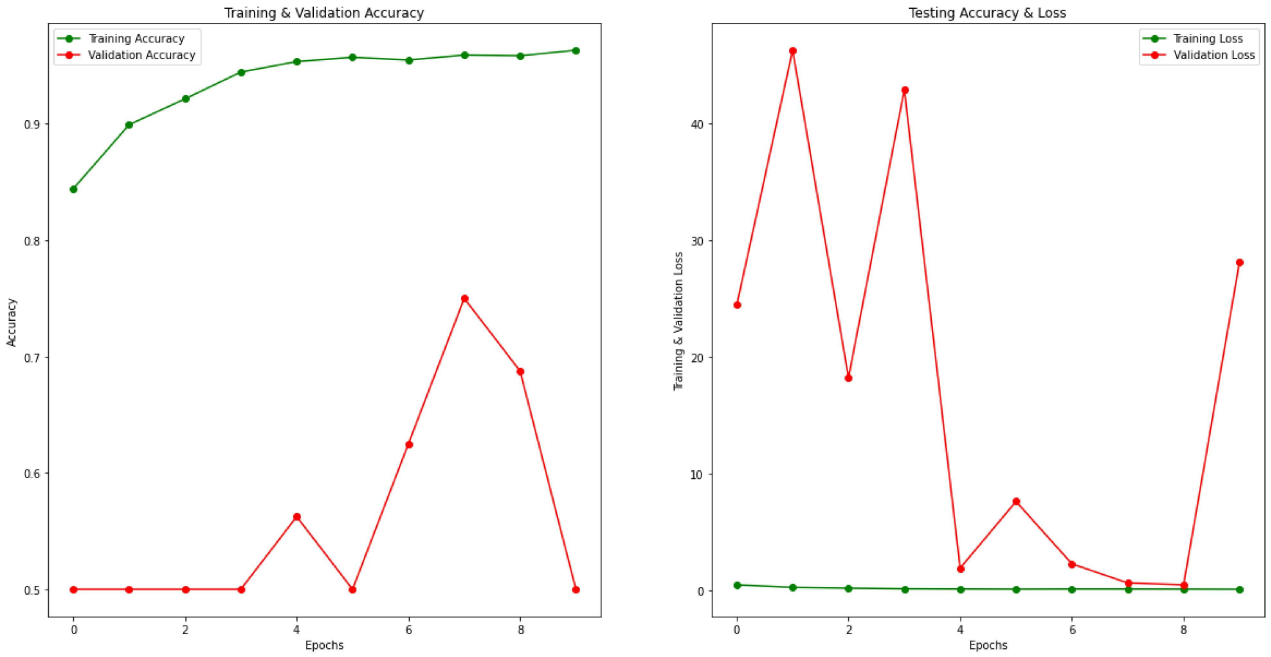
In [20]:

```
print("Loss of the model is - " , model.evaluate(x_test,y_test)[0])
print("Accuracy of the model is - " , model.evaluate(x_test,y_test)[1]*100 , "%")
```

```
20/20 [==============================] - 6s 280ms/step - loss: 6.2243 - accuracy: 0.6356
Loss of the model is -  6.224274635314941
20/20 [==============================] - 5s 272ms/step - loss: 6.2243 - accuracy: 0.6356
Accuracy of the model is -  63.56340050697327 %
```

In [21]:

```
epochs = list(range(10))
fig , ax = plt.subplots(1,2)
train_acc = history.history['accuracy']
train_loss = history.history['loss']
val_acc = history.history['val_accuracy']
val_loss = history.history['val_loss']
fig.set_size_inches(20,10)

ax[0].plot(epochs , train_acc , 'go-' , label = 'Training Accuracy')
ax[0].plot(epochs , val_acc , 'ro-' , label = 'Validation Accuracy')
ax[0].set_title('Training & Validation Accuracy')
ax[0].legend()
ax[0].set_xlabel("Epochs")
ax[0].set_ylabel("Accuracy")

ax[1].plot(epochs , train_loss , 'g-o' , label = 'Training Loss')
ax[1].plot(epochs , val_loss , 'r-o' , label = 'Validation Loss')
ax[1].set_title('Testing Accuracy & Loss')
ax[1].legend()
ax[1].set_xlabel("Epochs")
ax[1].set_ylabel("Training & Validation Loss")
plt.show()
```



In [22]:

```
predictions = model.predict(x_test)
for i in range(len(predictions)):
    predictions[i] = 1 if predictions[i]>0.5 else 0
```

```
20/20 [==============================] - 6s 264ms/step
```

In [23]:

```
print(classification_report(y_test,
                           predictions,
                           target_names = ['ARDS (Class 0)','Normal (Class 1)']))
```

```
                precision    recall  f1-score   support

  ARDS (Class 0)       0.63      1.00      0.77       389
Normal (Class 1)       1.00      0.03      0.06       234

        accuracy                           0.64       623
       macro avg       0.82      0.51      0.42       623
    weighted avg       0.77      0.64      0.51       623
```

In [24]:

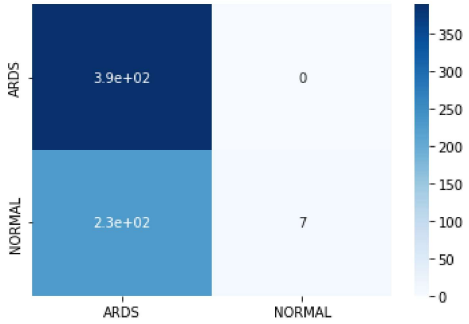
```
cm = confusion_matrix(y_test,predictions)
cm = pd.DataFrame(cm , index = ['0','1'] , columns = ['0','1'])
cm
```

Out[24]:

	0	1
0	389	0
1	227	7

In [25]:

```
sns.heatmap(cm, cmap="Blues", annot=True, xticklabels = labels,yticklabels = labels)
plt.show()
```



In [27]:

```
def ARDSPrediction(img):
    img = np.array(img)/255
    img = img.reshape(-1, 150, 150, 1)
    isARDS = model.predict(img)[0]
    imgClass = "Normal" if isARDS<0.5 else "ARDS"
    return imgClass
```

In [28]:

```
pr = model.predict(x_test)
for i in range(len(pr)):
    if pr[i]>0.5:
        pr[i]=1
    else:
        pr[i]=0
```

```
20/20 [==============================] - 5s 262ms/step
```

In [29]:

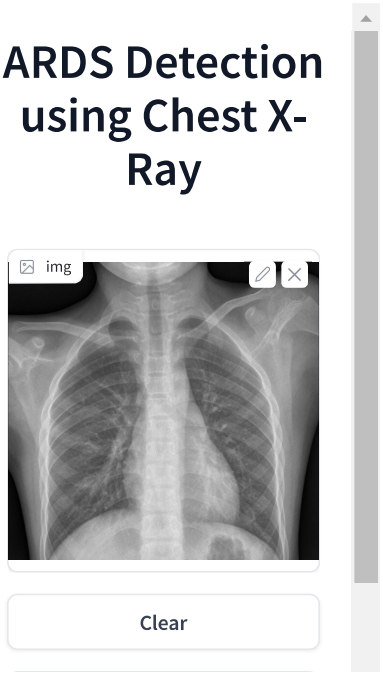
```
img = gradio.inputs.Image(shape=(150, 150))
label = gradio.outputs.Label(num_top_classes=1)
```

```
C:\Users\paturu vineetha\anaconda3\lib\site-packages\gradio\inputs.py:256: UserWarning: Usage of gradio.inputs is deprecate
d, and will not be supported in the future, please import your component from gradio.components
  warnings.warn(
C:\Users\paturu vineetha\anaconda3\lib\site-packages\gradio\deprecation.py:40: UserWarning: `optional` parameter is depreca
ted, and it has no effect
  warnings.warn(value)
C:\Users\paturu vineetha\anaconda3\lib\site-packages\gradio\outputs.py:196: UserWarning: Usage of gradio.outputs is depreca
ted, and will not be supported in the future, please import your components from gradio.components
  warnings.warn(
C:\Users\paturu vineetha\anaconda3\lib\site-packages\gradio\deprecation.py:40: UserWarning: The 'type' parameter has been d
eprecated. Use the Number component instead.
  warnings.warn(value)
```

In [*]:

```
interface = gradio.Interface(fn = ARDSPrediction,
                             title = "ARDS Detection using Chest X-Ray",
                             inputs = img,
                             outputs = label,
                             interpretation = "default")
interface.launch(debug=True, share=True)
```

```
Running on local URL:  http://127.0.0.1:7862 (http://127.0.0.1:7862)
Running on public URL: https://43aee444-eef2-4418.gradio.live (https://43aee444-eef2-4418.gradio.live)

This share link expires in 72 hours. For free permanent hosting and GPU upgrades (NEW!), check out Spaces: https://huggingf
ace.co/spaces (https://huggingface.co/spaces)
```



In [ ]:

In [ ]:

In [ ]:

In [ ]: