Chest X-Ray (Pneumonia): Image Classification w/Convolutional Neural Networks and Transfer Learning

Introduction

Pneumonia is a severe inflammatory condition of the lungs, primarily affecting the alveoli, and is a leading cause of death worldwide, especially in young children. Early and accurate diagnosis is crucial for effective treatment. A common diagnostic tool is the chest X-ray. However, visual interpretation of X-rays can be challenging, time-consuming, and prone to human error. This project explores the use of Convolutional Neural Networks (CNNs) and transfer learning to automate the detection of pneumonia from chest X-ray images, aiming to provide a rapid and objective screening tool to assist radiologists and clinicians.

Kaggle Dataset Link: https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia/data

Dataset Information

The dataset contains 5,856 validated Chest X-Ray images. The images are split into a training set and a testing set of independent patients. Images are labeled as (disease:NORMAL/BACTERIA/VIRUS)-(randomized patient ID)-(image number of a patient).

Chest X-ray images (anterior-posterior) were selected from retrospective cohorts of pediatric patients of one to five years old from Guangzhou Women and Children's Medical Center, Guangzhou. All chest X-ray imaging was performed as part of patients' routine clinical care.

For the analysis of chest x-ray images, all chest radiographs were initially screened for quality control by removing all low quality or unreadable scans. The diagnoses for the images were then graded by two expert physicians before being cleared for training the AI system. In order to account for any grading errors, the evaluation set was also checked by a third expert.

The dataset is organised organized into training and testing folders. Training consist of 5232 images while the testing consist of 624 images.

Importing Packages and Dataset

In [218…
```
import pandas as pd
import matplotlib as mlp
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
%matplotlib inline

pd.options.display.max_colwidth = 100

import random
import os
from IPython.display import Image, display
import matplotlib.cm as cm

from numpy.random import seed
seed(42)

random.seed(42)
os.environ['PYTHONHASHSEED'] = str(42)
os.environ['TF_DETERMINISTIC_OPS'] = '1'

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras import callbacks
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Input
from tensorflow.keras.applications.vgg16 import preprocess_input as vgg_preprocess

import glob
import cv2
from collections import Counter

from tensorflow.random import set_seed
set_seed(42)

import warnings
warnings.filterwarnings('ignore')

from sklearn.metrics import confusion_matrix,\
      roc_auc_score, ConfusionMatrixDisplay, roc_curve
```

[8 [ ]:
```
# Checking if TensorFlow is in GPU mode
gpus = tf.config.experimental.list_physical_devices('GPU')
if gpus:
    try:
        # Currently, memory growth needs to be the same across GPUs
        for gpu in gpus:
            print(gpu)
            tf.config.experimental.set_memory_growth(gpu, True)
        print("TensorFlow is using the GPU.")
    except RuntimeError as e:
        print(e)
else:
    print("TensorFlow is not using the GPU. Check your TensorFlow installation.")
```

```
PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')
TensorFlow is using the GPU.
```

Data

[8 [3]:
```
IMG_SIZE = 224
BATCH = 32
SEED = 42
```

[8 [4]:
```
main_path = "../Data/chest_xray"

train_path = os.path.join(main_path,"train")
test_path=os.path.join(main_path,"test")

train_normal = glob.glob(train_path+"/NORMAL/*.jpeg")
train_pneumonia = glob.glob(train_path+"/PNEUMONIA/*.jpeg")
```

```
test_normal = glob.glob(test_path+"/NORMAL/*.jpeg")
test_pneumonia = glob.glob(test_path+"/PNEUMONIA/*.jpeg")
```

In [5]:
```
train_list = [x for x in train_normal]
train_list.extend([x for x in train_pneumonia])

df_train = pd.DataFrame(np.concatenate([['Normal']*len(train_normal) , ['Pneumonia']*len(train_pneumonia)]), columns = ['class'])
df_train['image'] = [x for x in train_list]

test_list = [x for x in test_normal]
test_list.extend([x for x in test_pneumonia])

df_test = pd.DataFrame(np.concatenate([['Normal']*len(test_normal) , ['Pneumonia']*len(test_pneumonia)]), columns = ['class'])
df_test['image'] = [x for x in test_list]
```

In [6]:
```
df_train
```

Out[6]:

	class	image
0	Normal	../Data/chest_xray/train/NORMAL/NORMAL2-IM-0927-0001.jpeg
1	Normal	../Data/chest_xray/train/NORMAL/NORMAL2-IM-1056-0001.jpeg
2	Normal	../Data/chest_xray/train/NORMAL/IM-0427-0001.jpeg
3	Normal	../Data/chest_xray/train/NORMAL/NORMAL2-IM-1260-0001.jpeg
4	Normal	../Data/chest_xray/train/NORMAL/IM-0656-0001-0001.jpeg
...	...	...
5227	Pneumonia	../Data/chest_xray/train/PNEUMONIA/person142_virus_288.jpeg
5228	Pneumonia	../Data/chest_xray/train/PNEUMONIA/person364_bacteria_1659.jpeg
5229	Pneumonia	../Data/chest_xray/train/PNEUMONIA/person1323_virus_2283.jpeg
5230	Pneumonia	../Data/chest_xray/train/PNEUMONIA/person772_virus_1401.jpeg
5231	Pneumonia	../Data/chest_xray/train/PNEUMONIA/person501_virus_1010.jpeg

5232 rows × 2 columns

In [7]:
```
df_test
```

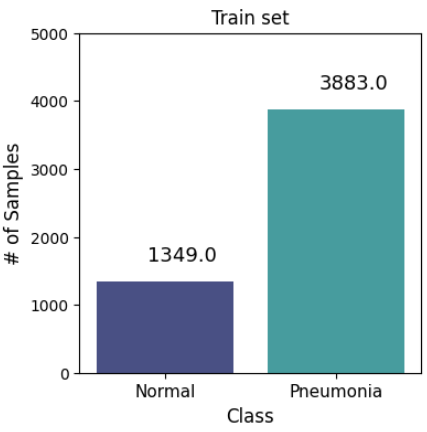
Out[7]:

	class	image
0	Normal	../Data/chest_xray/test/NORMAL/IM-0031-0001.jpeg
1	Normal	../Data/chest_xray/test/NORMAL/IM-0025-0001.jpeg
2	Normal	../Data/chest_xray/test/NORMAL/NORMAL2-IM-0272-0001.jpeg
3	Normal	../Data/chest_xray/test/NORMAL/NORMAL2-IM-0102-0001.jpeg
4	Normal	../Data/chest_xray/test/NORMAL/NORMAL2-IM-0229-0001.jpeg
...	...	...
619	Pneumonia	../Data/chest_xray/test/PNEUMONIA/person120_bacteria_572.jpeg
620	Pneumonia	../Data/chest_xray/test/PNEUMONIA/person171_bacteria_826.jpeg
621	Pneumonia	../Data/chest_xray/test/PNEUMONIA/person109_bacteria_512.jpeg
622	Pneumonia	../Data/chest_xray/test/PNEUMONIA/person83_bacteria_410.jpeg
623	Pneumonia	../Data/chest_xray/test/PNEUMONIA/person112_bacteria_538.jpeg

624 rows × 2 columns

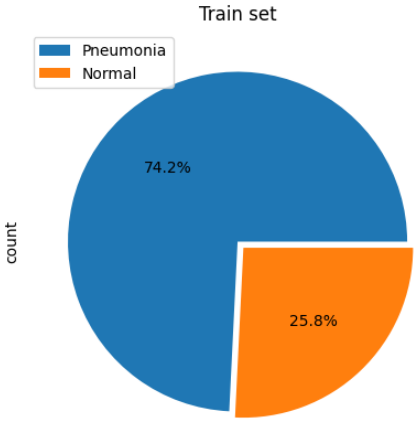
Data Exploration

In [8]:
```
plt.figure(figsize=(4,4))

ax = sns.countplot(x='class', data=df_train, palette="mako")

plt.xlabel("Class", fontsize= 12)
plt.ylabel("# of Samples", fontsize= 12)
plt.ylim(0,5000)
plt.title('Train set')
plt.xticks([0,1], ['Normal', 'Pneumonia'], fontsize = 11)

for p in ax.patches:
    ax.annotate((p.get_height()), (p.get_x()+0.30, p.get_height()+300), fontsize = 13)

plt.show()
```

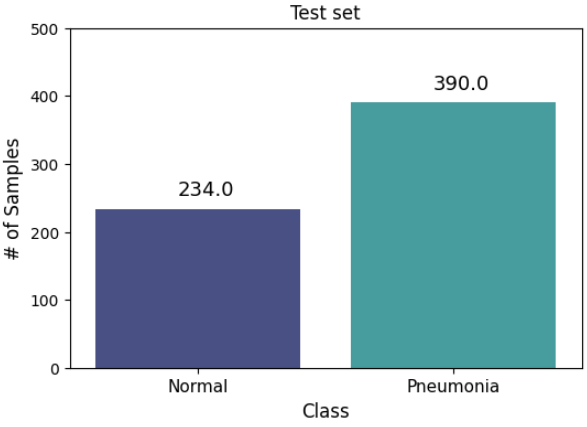


In [9]:
```
plt.figure(figsize=(5,5))

df_train['class'].value_counts().plot(kind='pie',labels = ['',''], autopct='%1.1f%%', explode = [0,0.05])
plt.title('Train set')
plt.legend(labels=['Pneumonia', 'Normal'])
```

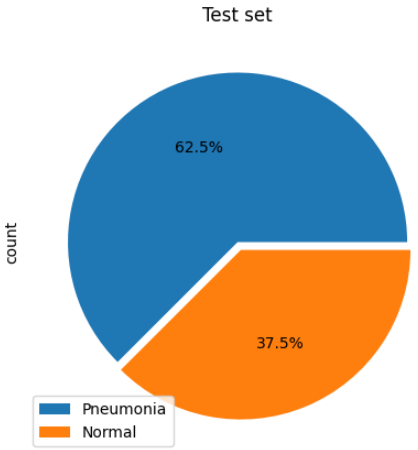
```
plt.show()
```



In [10]:
```
plt.figure(figsize=(6,4))

ax = sns.countplot(x='class', data=df_test, palette="mako")

plt.xlabel("Class", fontsize= 12)
plt.ylabel("# of Samples", fontsize= 12)
plt.ylim(0,500)
plt.title('Test set')
plt.xticks([0,1], ['Normal', 'Pneumonia'], fontsize = 11)

for p in ax.patches:
    ax.annotate((p.get_height()), (p.get_x()+0.32, p.get_height()+20), fontsize = 13)

plt.show()
```



In [11]:
```
plt.figure(figsize=(7,5))

df_test['class'].value_counts().plot(kind='pie',labels = ['',''], autopct='%1.1f%%', explode = [0,0.05])
plt.title('Test set')
plt.legend(labels=['Pneumonia', 'Normal'])
plt.show()
```



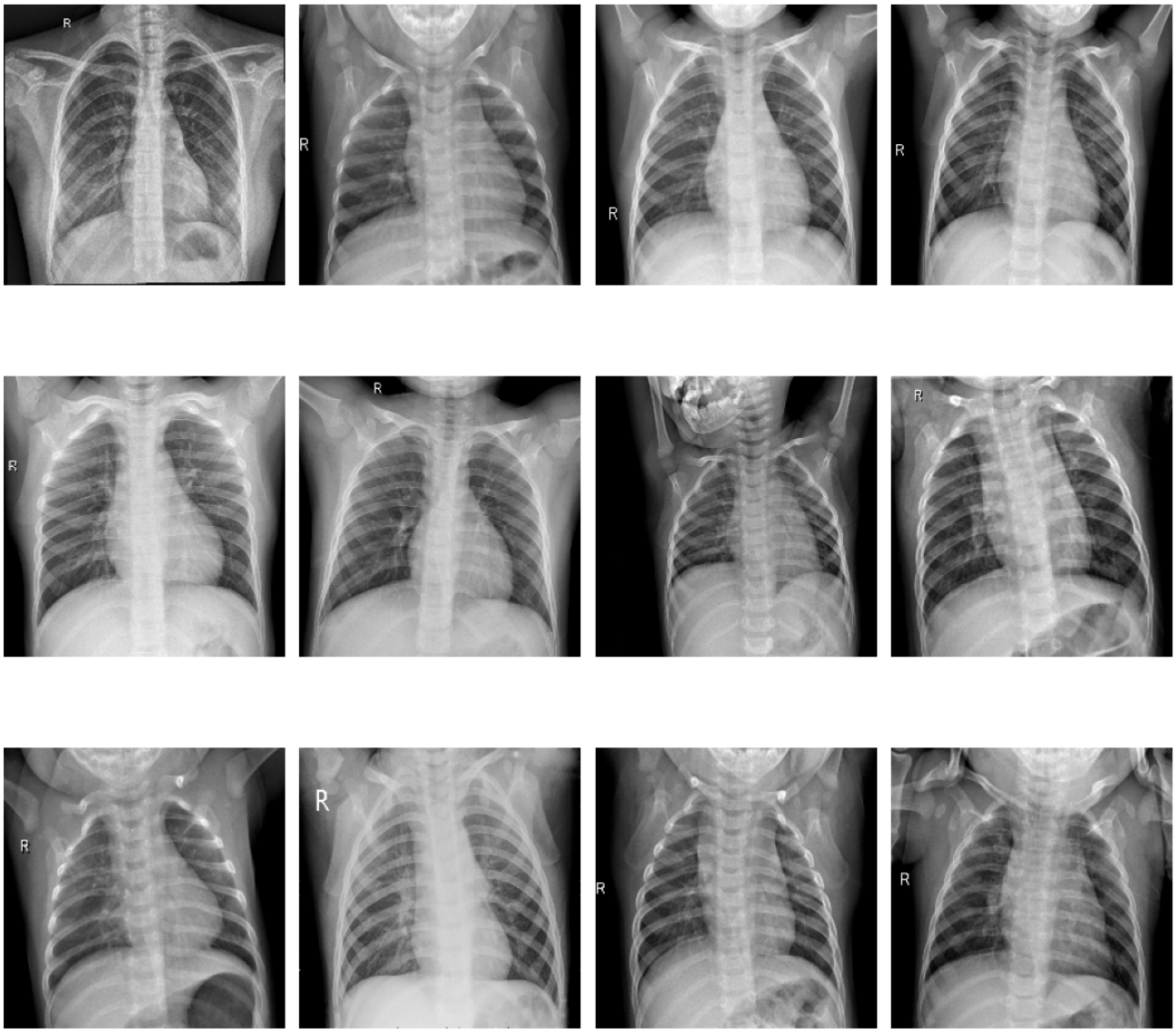
The distributions from these datasets are a little different from each other. Both are slightly imbalanced, having more samples from the positive class (Pneumonia), with the training set being a little more imbalanced.

Before we move on to the next section, we will take a look at a few examples from each dataset.

In [12]:
```
print('Train Set - Normal')

plt.figure(figsize=(12,12))

for i in range(0, 12):
    plt.subplot(3,4,i + 1)
    img = cv2.imread(train_normal[i])
    img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))
    plt.imshow(img)
    plt.axis("off")
```

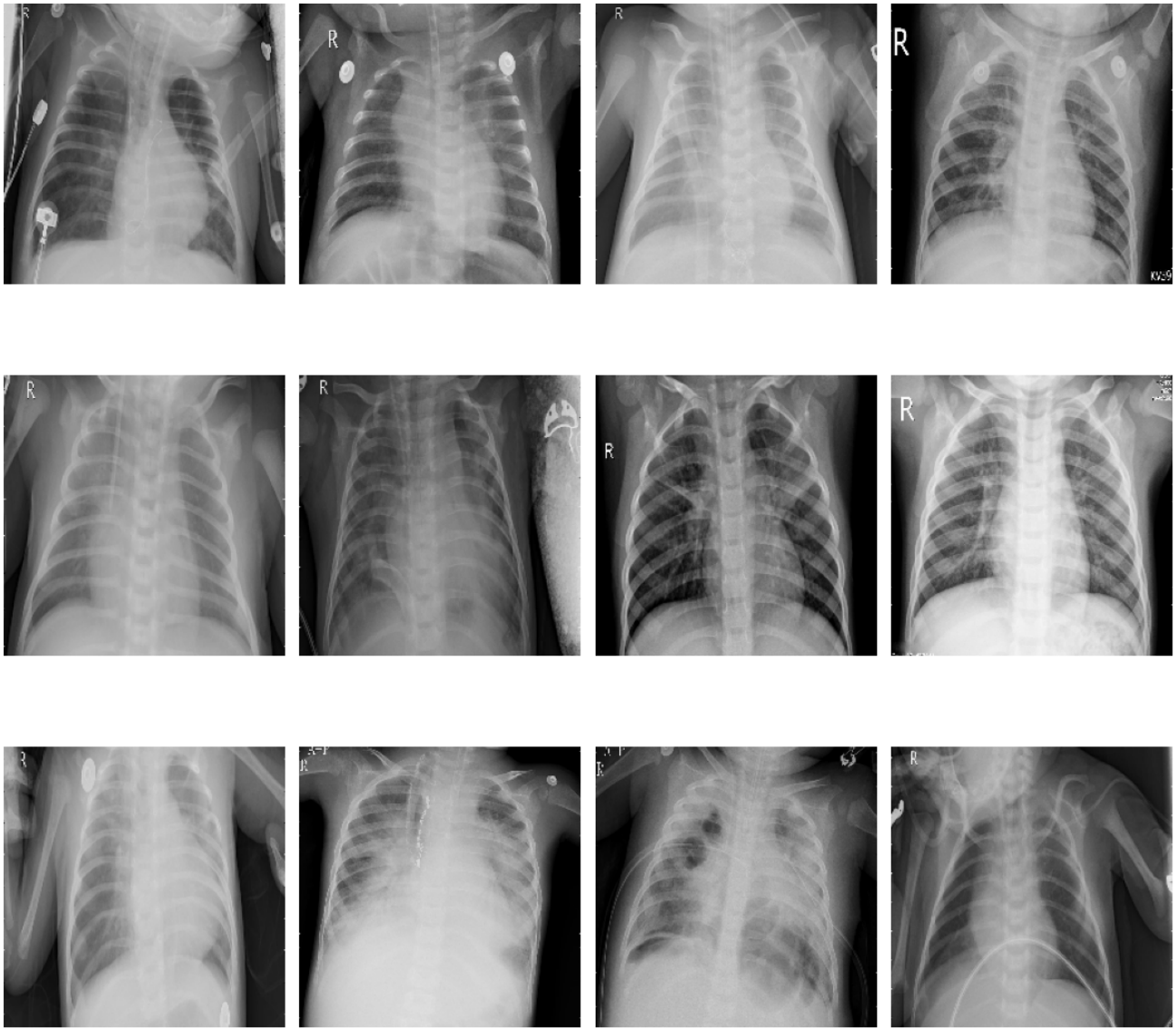
```
plt.tight_layout()

plt.show()
```

```
Train Set - Normal
```



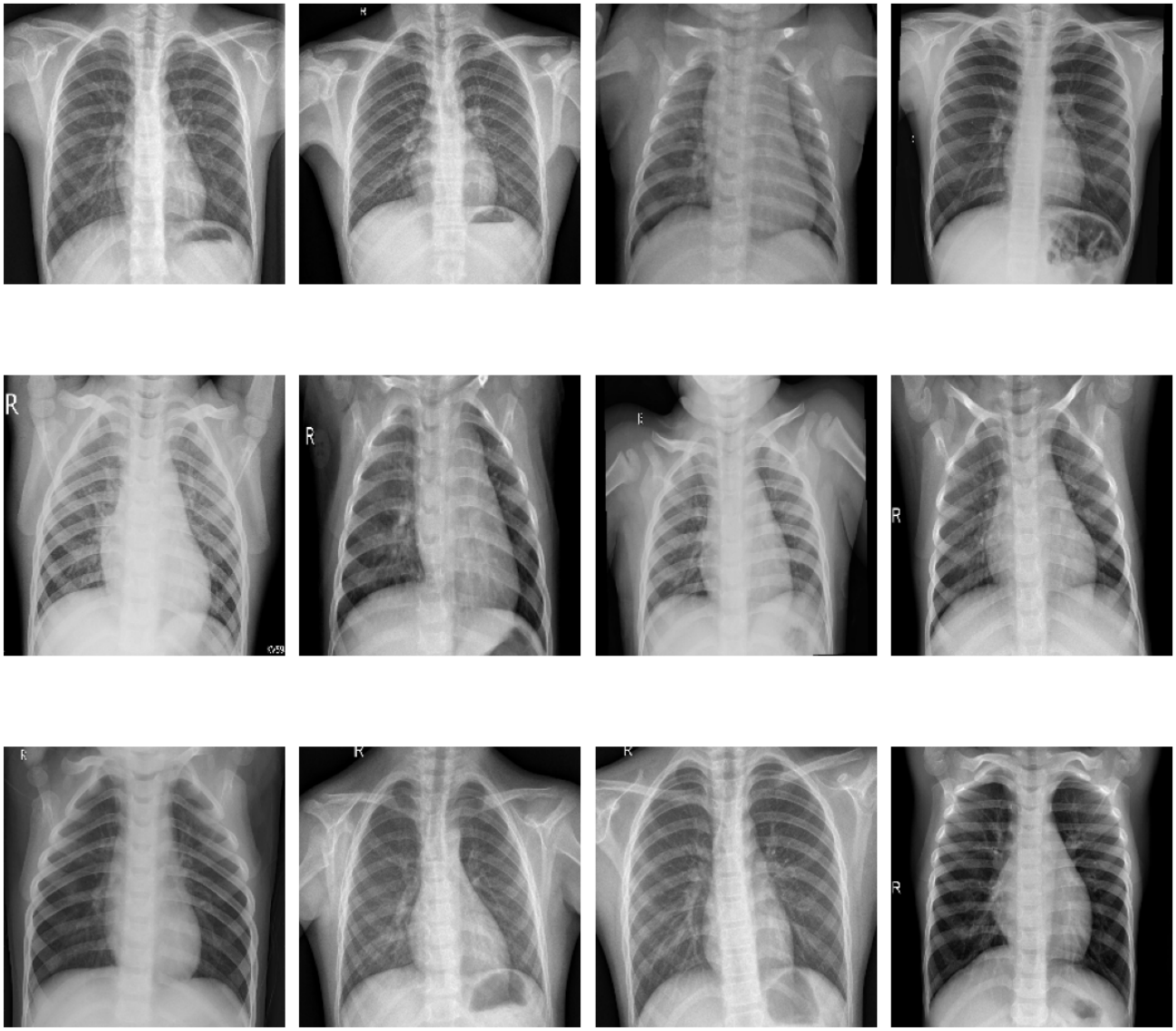
In [13]:
```
print('Train Set - Pneumonia')

plt.figure(figsize=(12,12))

for i in range(0, 12):
    plt.subplot(3,4,i + 1)
    img = cv2.imread(train_pneumonia[i])
    img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))
    plt.imshow(img)
    plt.axis("off")

plt.tight_layout()

plt.show()
```

```
Train Set - Pneumonia
```



In [14]:
```
print('Test Set - Normal')

plt.figure(figsize=(12,12))

for i in range(0, 12):
    plt.subplot(3,4,i + 1)
    img = cv2.imread(test_normal[i])
    img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))
    plt.imshow(img)
    plt.axis("off")

plt.tight_layout()

plt.show()
```

```
Test Set - Normal
```



In [115]:
```
print('Test Set - Pneumonia')

plt.figure(figsize=(12,12))

for i in range(0, 12):
    plt.subplot(3,4,i + 1)
    img = cv2.imread(test_pneumonia[i])
    img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))
    plt.imshow(img)
    plt.axis("off")

plt.tight_layout()

plt.show()
```

```
Test Set - Pneumonia
```