Generative AI Project

Medical Image Synthesis using GANs for Pulmonary Chest X-rays

BY HARISANTHOSH
au211521243063

INTRODUCTION

"Developing an efficient and accurate method for medical image synthesis using Generative Adversarial Networks (GANs) to generate synthetic Pulmonary Chest X-rays, aiming to address the scarcity of labeled data for training deep learning models in pulmonary diagnostics and enhance the robustness and generalization of pulmonary disease detection algorithms."

Objective:

Develop a deep learning model using Generative Adversarial Networks (GANs) to synthesize realistic pulmonary chest X-rays for medical training and diagnostic purposes.

ALGORITHMS FOR PULMONARY CHEST X RAYS

Data Collection and Preprocessing:

Gather a dataset of pulmonary chest X-ray images.
Preprocess the images (e.g., resizing, normalization) to ensure consistency and improve training efficiency.
Generator Network Design:

Design a generator network architecture suitable for synthesizing realistic chest X-ray images.
Typically, a convolutional neural network (CNN) is used as the generator, with upsampling layers to generate high-resolution images.
Discriminator Network Design:

Design a discriminator network to distinguish between real and synthetic chest X-ray images.
CNN architecture is commonly employed for the discriminator, with downsampling layers to extract features.
Training Process:

Initialize the generator and discriminator networks.
Alternately train the generator and discriminator networks in a minimax game:
Generator: Generate synthetic chest X-ray images from random noise or an initial input.
Discriminator: Distinguish between real and synthetic chest X-ray images.
Update the weights of both networks based on the adversarial loss function.
Repeat the training process iteratively until convergence or a stopping criterion is met.

code implementation

Step 1: Importing Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')

import PIL
import tensorflow as tf
from keras.models import Sequential
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Step 2: Loading Data

```
image = PIL.Image.open('/kaggle/input/chest-xray-
pneumonia/chest_xray/train/PNEUMONIA/person1000_bacteria_2931.jpeg')
image
```

Step 3: Exploring Data

```
rain_dir = '/kaggle/input/chest-xray-pneumonia/chest_xray/train'
training_generator = ImageDataGenerator(rescale = 1/255)
train_data = training_generator.flow_from_directory(train_dir, target_size=(120,120), batch_size=8,
class_mode='binary')
```

Step 4: Data Preprocessing

```
valid_dir = '/kaggle/input/chest-xray-pneumonia/chest_xray/val'
validation_generator = ImageDataGenerator(rescale = 1/255)
valid_data = validation_generator.flow_from_directory(valid_dir, target_size=(120,120), batch_size=8,
class_mode='binary')
```

Step 5: Train-Test Split

```
test_dir = '/kaggle/input/chest-xray-pneumonia/chest_xray/test'
testing_generator = ImageDataGenerator(rescale = 1/255)
test_data = testing_generator.flow_from_directory(test_dir, target_size=(120,120),
batch_size=8, class_mode='binary')
```

Step 6: Model Training

```
model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(32, (3,3), input_shape=(120,120,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(256, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(512, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(256, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
```

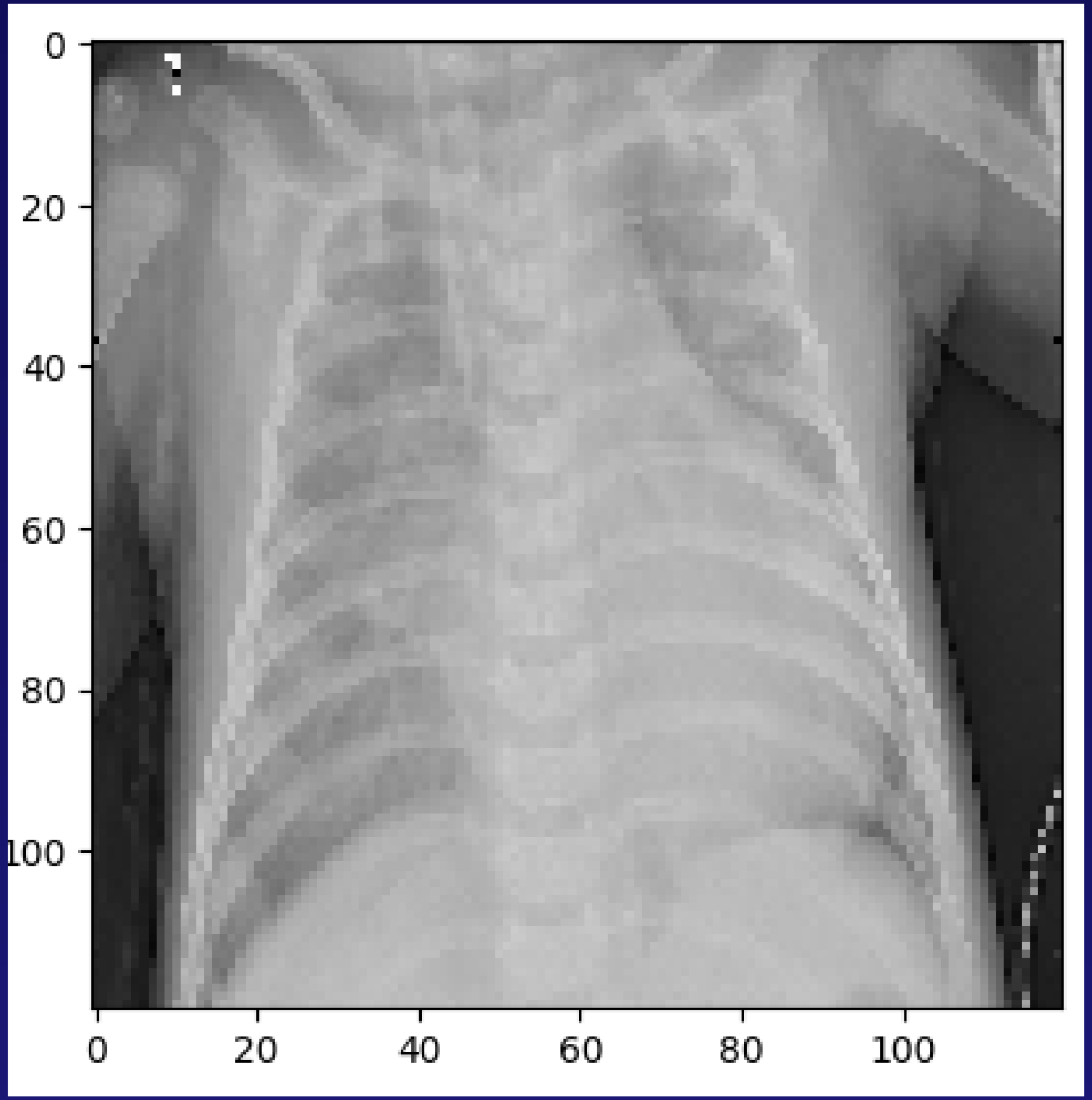
Step 8: Making Predictions

```
predictions = model.predict(test_data)
```

```
batch_size = test_data.batch_size
num_batches = 1

for i in range(num_batches):
    x, y = next(test_data)
    predictions = model.predict(x)

    for j in range(min(batch_size, 8)):
        image = x[j]
        plt.imshow(image)
        plt.show()
        print('The probability of Pneumonia is: ',
predictions[j]
```



Challenges in Implementing PULMONARY CHEST X RAYS:

Image Quality and Interpretability: Ensuring high-quality images free from artifacts is crucial for accurate diagnosis, but factors like patient positioning and technical limitations can hinder image clarity, impacting interpretation.

Radiation Exposure: Although chest X-rays involve lower radiation doses than CT scans, managing radiation exposure levels, especially in vulnerable populations, remains a concern for patient safety.

Variability in Anatomy and Pathology: The human chest exhibits diverse anatomical structures and pathological conditions, making it challenging to develop algorithms that can accurately detect abnormalities across different patient populations and disease presentations.

Data Annotation and Labeling: Annotating chest X-ray images with accurate labels for training machine learning models is labor-intensive and requires expertise. Subjective findings and rare conditions further complicate the labeling process, potentially leading to inconsistencies in the dataset.