ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

TITLE: PNEUMONIA DETECTION USING MACHINE LEARNING

Team Number:

Team Members:

2320040050 - ALURU DOONDI BALA SARASWATHI

2320040115 - SHRIYA KANDAKATLA

Abstract:

This project focuses on the development of a Pneumonia Detection system using deep learning, specifically leveraging the VGG16 architecture pre-trained on ImageNet. The system is designed to classify chest X-ray images into three categories: Bacterial Pneumonia, Viral Pneumonia, and Normal. By using a CNN, the model processes X-ray images, utilizing image preprocessing techniques such as rescaling, shear transformation, and zoom augmentation to enhance the training data. The model is fine-tuned to improve accuracy and prevent overfitting, and is then deployed as a web application using Streamlit. The application allows users to input image URLs for real-time classification, providing an accessible and efficient tool for pneumonia detection.

Problem Implementation Overview:

1. Data Preprocessing:

Approach: The data preprocessing involves reading configuration settings from a YAML file, followed by loading and augmenting the training and test datasets using ImageDataGenerator. The augmentation techniques include rescaling pixel values, applying shear transformations, and zooming. The preprocessed data is then stored along with metadata such as image shape, number of batches, and class indices.

2. Model Training:

Approach: The model training utilizes the VGG16 architecture, pre-trained on ImageNet, with its convolutional layers frozen to retain pre-learned weights. A custom dense layer is added for classification. The model is compiled with categorical cross-entropy loss, an Adam optimizer, and accuracy metrics. It is then trained on the augmented dataset, and the training process is logged, including time taken, loss, and accuracy. The trained model is saved in the specified directory for future use.

3. Model Evaluation:

Approach: The evaluation script loads the trained model and tests it on the preprocessed test dataset. It predicts class labels for the test set and computes evaluation metrics including precision, recall, F1-score, and a confusion matrix. These metrics are logged and stored for future reference.

4. Web Application:

Approach: The deployment of the trained model is achieved through a Streamlit web application. Users can input a URl of a chest X-ray image, which is then processed and classified by the model. The predicted class is displayed along with the corresponding image. The application leverages the cached model for efficient performance.

5. Project Setup:

Approach: This script initializes the project directory structure, creating necessary folders and files such as training and test data directories, model storage, and reporting folders. It also generates placeholders for essential files like params.yaml and requirements.txt to ensure a well-organized project setup.