Pneumonia Detection in Chest X-Rays: An Analysis of Convolutional Neural Networks

BY KARTHIK REDDY J
Reg no: 23MAS10008
Under the Guidance of: Dr. Virendra Singh Kushwah

METHODOLOGY

- **Develop a Convolutional Neural Network (CNN) Model:** Create CNN- based deep learning model designed specifically for the detection of pneumonia in chest X-ray images.
- **Data Collection and Preparation:** Collect a comprehensive dataset of chest X-ray images, including both pneumonia-positive and pneumonia-negative cases. Preprocess and augment the data to ensure suitability for training and testing the CNN model.
- **Model Training and Optimization:** Train the CNN model on the dataset, optimizing hyperparameters and architecture to achieve the highest detection accuracy while minimizing false positives and false negatives.
- **Cross-validation:** Employ cross-validation techniques to assess the robustness and generalizability.

Problem Statement

- Several preprocessing techniques can be used to resolve the class imbalance problems.
- The training dataset is often expanded via data augmentation, resulting in false representations of the original images to avoid overfitting.
- The quality of the chest X-ray images can vary significantly, which may affect model performance.

DATASET DESCRIPTION

DATASET SOURCES:

Dataset used for this research was collected from Kaggle : Public Domain by ALIF RAHMAN.

Link to Dataset:

https://www.kaggle.com/datasets/alifrahman/chestxraydataset

Content:

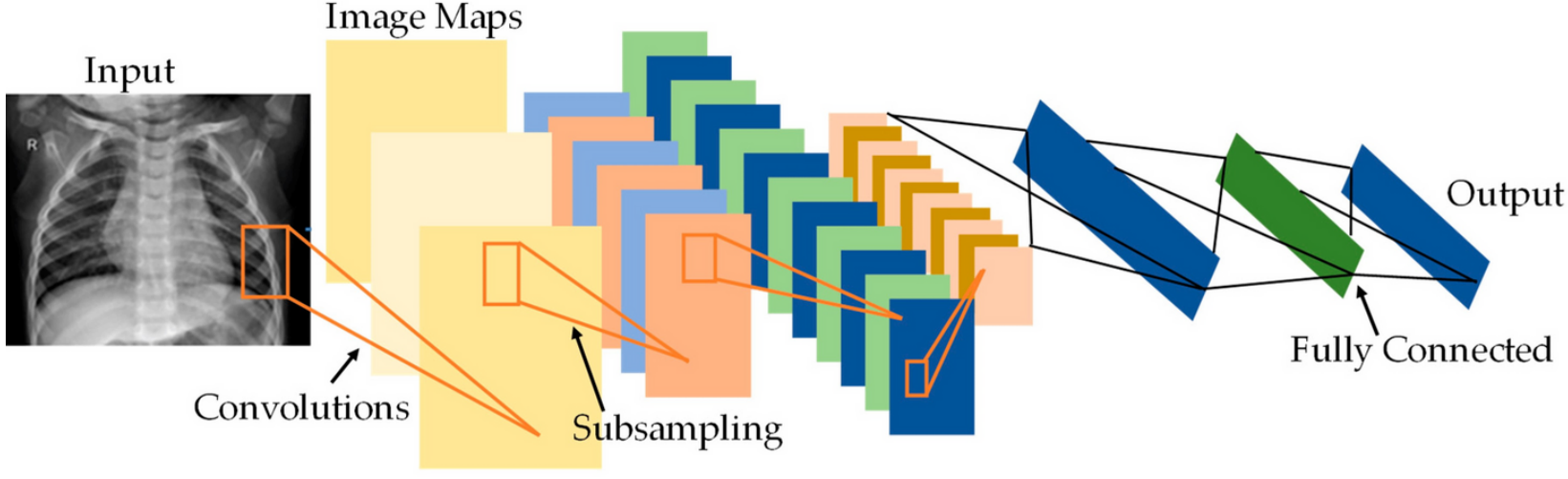
This dataset mainly consists of the chest X-ray images of Normal and Pneumonia affected patients. There is a total of 5840 chest X-ray images. Each of them has two sub-folders labeled as NORMAL and PNEUMONIA. To acquire this dataset, we can directly download it from the provided links. The process may consume some time due to the dataset's substantial size, approximately 1GB in total.

LITERATURE SURVEY

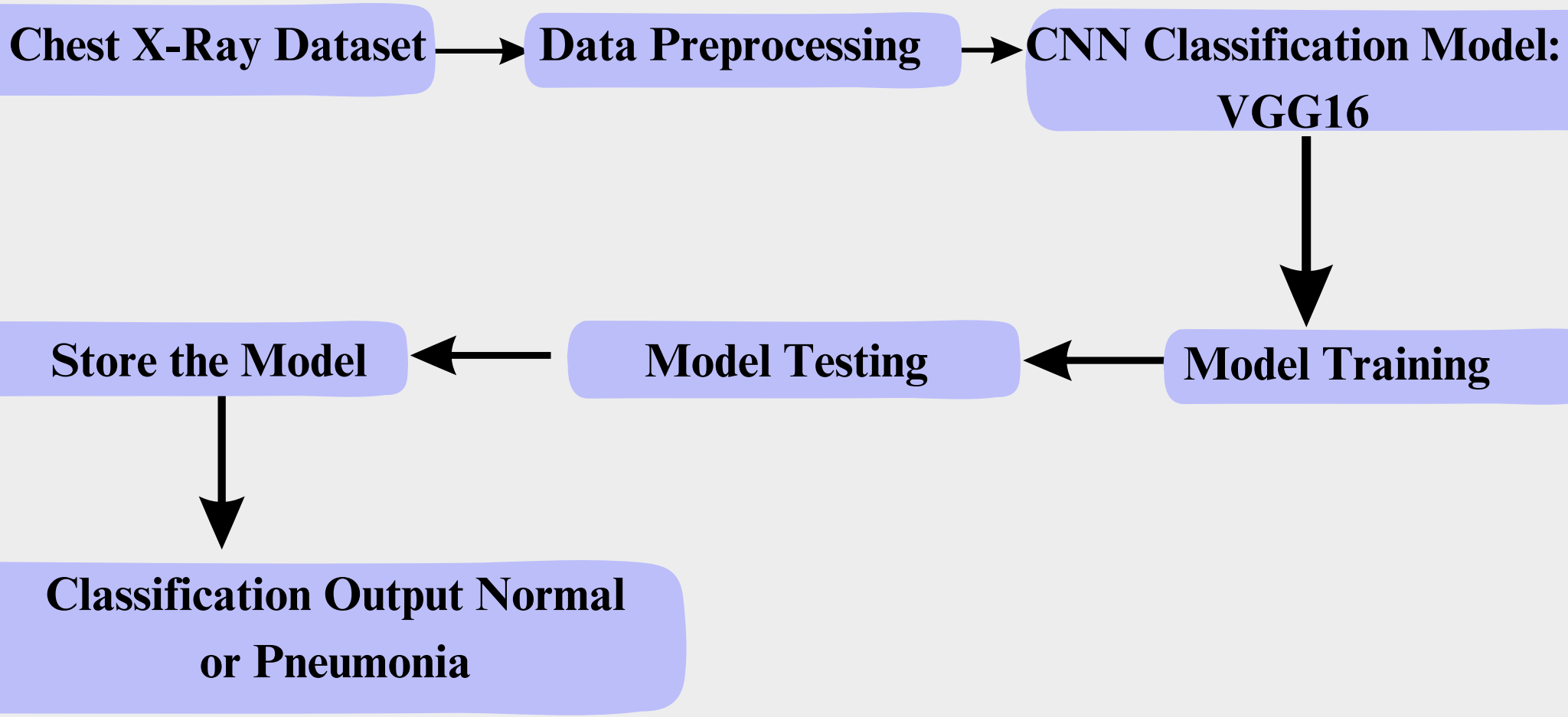
TITLE	METHODOLOGY	INFERENCE
Efficient Pneumonia Detection in Chest Xray Images Using Deep Transfer Learning	Deep Learning, Pretrained Model	An accuracy of 98.857%, and the efficiency of the proposed model.
An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare	Deep Learning, Transfer Learning	• A traditional CNN Model has been Used for feature Extraction and classification of pneumonia • Training accuracy 0.95 and validation accuracy of 0.9373.
Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection Using Chest X-ray	Pretrained Model	Pretrained model exhibits an excellent performance in classifying pneumonia by effectively training itself from a comparatively collection of data set.

CNN(Convolutional Neural Network)

- In deep learning, a convolutional neural network (CNN) is a class of deep neural networks, most commonly applied to analyze visual imagery.
- **Convolutional Layer** : Convolutional Layers will compute the output of nodes that are connected to local regions of the input matrix. These layers are made of many filters, which are defined by their width, height, and depth.
- **Pooling Layers:** After the detectives find patterns, the down-sampling strategy is applied to reduce the width and height of the output volume.
- **Fully Connected Layer:** The final layer is the fully connected layers which takes the high-level filtered images and translate them into categories with labels.



Proposed System



- To detect the pneumonia Image sets undergo some pre-processing steps, data augmentation, and then training using pre-trained algorithms.
- Our goal is to detect the pneumonia by analyzing radiograph with fast classification and better accuracy.

DATA PREPROCESSING

- **Train_datagen and Test_datagen: These are instances of the ImageDataGenerator class, responsible for data augmentation and normalization.**

- **They rescale pixel values to the range [0, 1] (normalization) and apply various transformations to the images to create additional training data. These transformations include random rotations, horizontal and vertical shifts, shearing, zooming, and horizontal flipping.**

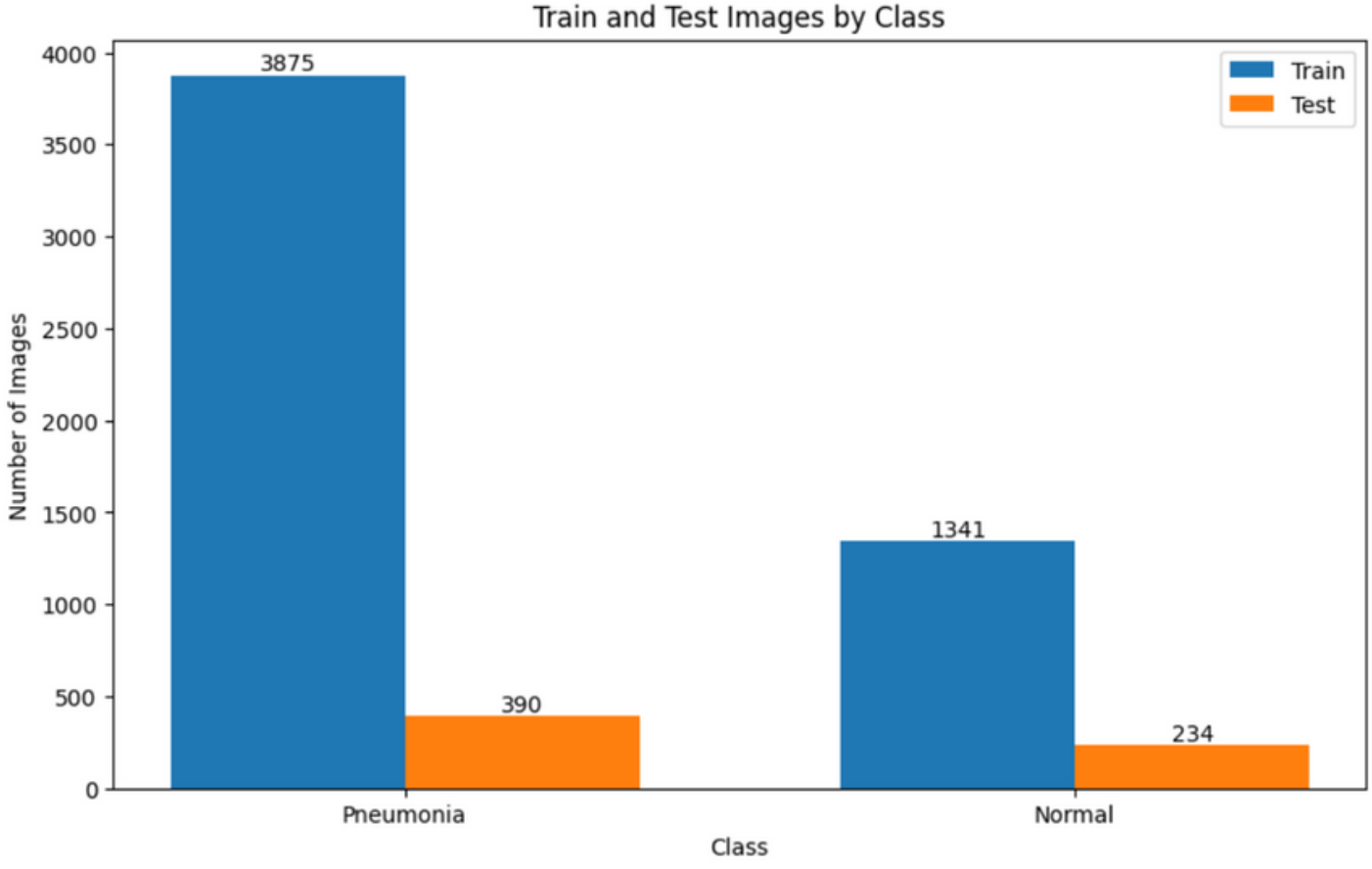
- **Train_generator and test_generator: These are created using flow_from_directory from the Image Data Generator instances. They load and preprocess images from directories containing training and testing data.**

- **Train_generator is used for model training, and test_generator is used for evaluation. The batch_size, target_size, and class_mode parameters are set for generating batches of data.**

```
# Data augmentation and normalization
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=15,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='binary'
)

test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='binary',
    shuffle=False
)
```

```
Found 5216 images belonging to 2 classes.
Found 624 images belonging to 2 classes.
```

DATA VISUALIZATION



No. Of Pneumonia Train Images: 3875

No. Of Pneumonia Test Images: 390

No. Of Normal Train Images: 1341

No. Of Normal Test Images: 234

DEFINING CNN

```
# Pre-trained model (VGG16)
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

# Fine-tuning the pre-trained model
model = Sequential()
model.add(base_model)
model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))

# Compile the model
model.compile(
    loss='binary_crossentropy',
    optimizer=Adam(learning_rate=learning_rate),
    metrics=['accuracy']
)
```

1. **It imports the VGG16 pre-trained model with weights from the ImageNet dataset, excluding the top classification layer.**
2. **A new neural network model is created and initialized as model.**
3. **The VGG16 model is added as the base, followed by a flattening layer, two fully connected layers with ReLU and dropout, and a final output layer with sigmoid activation for binary classification.**
4. **The model is compiled with binary cross-entropy loss, the Adam optimizer with a specified learning rate, and accuracy as the evaluation metric.**

MODEL TRAINING

```
# Training the model
history = model.fit(
    train_generator,
    epochs=epochs,
    validation_data=test_generator
)
```

```
Epoch 18/20
163/163 [==============================] - 132s 811ms/step - loss: 0.0422 - accuracy: 0.9831 - val_loss: 0.2110 - val_accuracy: 0.9423
Epoch 19/20
163/163 [==============================] - 137s 837ms/step - loss: 0.0461 - accuracy: 0.9827 - val_loss: 0.2588 - val_accuracy: 0.9359
Epoch 20/20
163/163 [==============================] - 131s 803ms/step - loss: 0.0529 - accuracy: 0.9797 - val_loss: 0.1823 - val_accuracy: 0.9487
```

1. **The model was trained using the train_generator with a specified number of epochs (training iterations). During training, the model's performance was evaluated using the test_generator as the validation data.**
2. **After training, the model achieved a validation accuracy of 94% and a validation loss of 0.1823, indicating its performance in correctly classifying pneumonia cases in the test dataset.**

DATA TESTING

```python
# Evaluate the model
test_loss, test_accuracy = model.evaluate(test_generator)
print()
print("====================================")
print()
print(f"Test Loss: {test_loss}")
print()
print(f"Test Accuracy: {test_accuracy*100:.2f}%")
```

```
624/624 [==============================] - 8s 13ms/step - loss: 0.1823 - accuracy: 0.9487

====================================

Test Loss: 0.18231043219566345

Test Accuracy: 94.87%
```

1. **The code evaluates the trained model using the test_generator and calculates the test loss and test accuracy, providing a quantitative measure of how well the model performs on a separate test dataset.**
2. **The evaluation results show that the model achieved a test accuracy of 94% and a test loss of 0.1823, confirming its effectiveness in correctly classifying pneumonia cases on unseen data.**

RESULT

```python
import numpy as np
from sklearn.metrics import classification_report, confusion_matrix
from tensorflow.keras.models import load_model

# Load the trained model
model = load_model('pneumonia_detection_model.h5')

# Predict on the test data
predictions = model.predict(test_generator)
predicted_classes = np.round(predictions)

# Get the true labels
true_classes = test_generator.classes

# Generate a classification report
class_names = list(test_generator.class_indices.keys())
report = classification_report(true_classes, predicted_classes, target_names=class_names)

# Generate a confusion matrix
confusion = confusion_matrix(true_classes, predicted_classes)

print("Classification Report:\n")
print(report)

print("\nConfusion Matrix:\n")
print(confusion)
```

```
20/20 [==============================] - 6s 295ms/step
Classification Report:

              precision    recall  f1-score   support

      NORMAL       0.93      0.89      0.91       234
   PNEUMONIA       0.94      0.96      0.95       390

    accuracy                           0.93       624
   macro avg       0.93      0.92      0.93       624
weighted avg       0.93      0.93      0.93       624


Confusion Matrix:

[[208  26]
 [ 16 374]]
```

1. **The "Classification Report" provides a summary of the model's performance on a test dataset. It shows precision, recall, and F1-score for two classes: "NORMAL" and "PNEUMONIA." The model achieved high accuracy, with F1-scores of 0.91 for "NORMAL" and 0.95 for "PNEUMONIA."**
2. **The "Confusion Matrix" illustrates the model's predictions. It indicates that the model made 208 correct predictions for "NORMAL" cases and 374 correct predictions for "PNEUMONIA" cases, but it also had 26 false negatives (misclassified as "NORMAL") and 16 false positives (misclassified as "PNEUMONIA"). Overall, the model's performance appears strong.**

Key Reference:

- **Efficient Pneumonia Detection in Chest Xray Images Using Deep Transfer Learning -** Mohammad Farukh Hashmi , Satyarth Katiyar , Avinash G Keskar, Neeraj Dhanraj Bokde and Zong Woo Geem - 19 June 2020
- **An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare -** Okeke Stephen, Mangal Sain, Uchenna Joseph Maduh, and Do-Un Jeong - 27 Mar 2019.
- **Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection Using Chest X-ray -** Tawsifur Rahman,Muhammad E. H. Chowdhury , Amith Khandakar - 6 May 2020

Thank You