**AUTOMATED DIAGNOSTIC CLASSIFICATION OF CHEST X-RAYS**

**Introduction**

The classification of medical images plays a crucial role in diagnosing various respiratory conditions, including COVID-19, pneumonia and tuberculosis. With the availability of a diverse dataset containing images representative of these respiratory conditions, there is an opportunity to develop a multiclass neural network using Keras for accurate and efficient classification. Furthermore, this technology has the potential to revolutionize the healthcare industry by improving diagnostic accuracy and efficiency.

**Statement of Problem**

In the field of medical imaging, the rapid and accurate diagnosis of thoracic diseases is critical for effective patient treatment and management. However, manual interpretation of chest X-rays, which are among the most commonly performed radiological procedures, can be time-consuming and subject to variability based on the radiologist’s experience and number of cases to be reviewed. With the incident COVID-19 pandemic and the high prevalence of respiratory diseases such as pneumonia and tuberculosis, healthcare systems are under immense pressure, necessitating tools that can assist in quick and reliable diagnosis.

This project seeks to address the challenge of automating the classification of chest X-rays into four categories which includes COVID-19, pneumonia, tuberculosis and in normal study using a Convolutional Neural Network (CNN) model. The solution aims to aid radiologists in screening processes by providing preliminary diagnostic suggestions, thus reducing the time to diagnosis and helping prioritize urgent cases. By leveraging deep learning algorithms and a substantial dataset of labeled X-ray images, the model endeavors to match the diagnostic ability of an experienced radiologist in identifying and distinguishing between these critical respiratory conditions.

The successful implementation of this model could significantly impact public health by facilitating the early detection of infectious diseases, thereby improving patient outcomes and easing the burden on healthcare providers. It also serves as a foundational step towards more sophisticated diagnostic systems capable of interpreting a wider array of pathologies in medical images.

**Objective**

The primary objective is to design and implement a multiclass neural network model using Keras to classify medical images into one of four categories: normal, COVID-19, pneumonia, and tuberculosis. By leveraging deep learning techniques, the model aims to accurately differentiate between different respiratory conditions based on image features thereby contributing to improved diagnostic accuracy and patient care.

**Methodology**

This involved choosing the appropriate architecture, including the number of layers, types of layers (e.g., dense, convolutional, recurrent), activation functions, and output layer configuration. A sequential CNN model was constructed using Keras, consisting of multiple convolutional layers followed by max-pooling layers, a flattening step, a fully connected layers and finally a softmax layer for multiclass classification.

**Data Preprocessing**

Data augmentation and preprocessing techniques like rotation, width and height shifting, shearing, zooming, and horizontal flipping were applied to the training data, enhancing its diversity.

flow\_from\_directory method was utilized to generate batches of image data from the directory structure.

The dataset is split into training, validation, and test sets. ImageDataGenerator is utilized for data augmentation and rescaling for the training set, while only rescaling is applied to the validation and test sets. Here is a link to the dataset in google drive: <https://drive.google.com/drive/folders/1DOLempWnrLdU9b3hBWaJHS7slzhhb7yp?usp=drive_link>

**Model Architecture**

The CNN model was defined using the sequential API and it comprises:

Four convolutional layers with increasing filter sizes (32, 64, 128, 128) and ReLU activation functions, each followed by a max-pooling layer to extract feature from images.

A flattening layer transitions to fully connected layers.

A dropout layer (50% rate) is used for regularization to prevent overfitting.

The final dense layer with a softmax activation function outputs the probability distribution over the four classes.

**Training**

Training was performed using the fit method, specifying the number of epochs and utilizing both training and validation data generators. The model was trained for five epochs using the Adam optimizer and categorical cross-entropy loss function. Precision, recall, and F1-score are included as metrics alongside accuracy.

**Performance Evaluation**

The model's performance is evaluated on a previously unseen test set. The resulting confusion matrix shows correct and incorrect predictions, where ideally, correct predictions align along the diagonal.

**Confusion Matrix**

**True Positives (TP):** The diagonal cells from the top left to the bottom right represent the number of true positive predictions for each class, where the predicted class corresponds to the true class. For example, the cell in the top left corner (190) indicates that there were 190 instances where the model correctly identified COVID-19. The cell in the third row and third column (389) represents the true positive predictions for pneumonia, which suggests that the model performs best in identifying this class.

**False Positives (FP):** For a given column, any cell not on the diagonal represents a false positive for the predicted class. For instance, the second cell in the first column (2) shows that there were 2 instances where normal cases were incorrectly predicted as COVID-19.

**False Negatives (FN):** Conversely, for a given row, any cell not on the diagonal indicates that the model failed to identify the actual class (false negatives). Taking the first row as an example, there were 4 instances of COVID-19 that the model incorrectly identified as normal.

**True Negatives (TN):** Although not explicitly shown in the confusion matrix, true negatives for a particular class can be interpreted as all the cells not in the class's specific row or column.

**Class-Specific Analysis**

**COVID-19:** The model identified COVID-19 with strong accuracy, but there is notable confusion with pneumonia cases, indicated by 16 instances each being misclassified.

**Normal:** The model shows a propensity to confuse normal instances as pneumonia, with a significant number of misclassifications (145). It suggests that the model's features or patterns learned for distinguishing normal from pneumonia are not adequately discriminative.

**Pneumonia:** The model demonstrates a strong ability to recognize pneumonia, with a high number of true positives (389). However, there's also a high rate of pneumonia cases misclassified as normal, which could indicate an overlap in the features or a need to refine the criteria for what distinguishes normal cases from pneumonia.

**Tuberculosis:**  The model has strong performance with (152) correct predictions but also shows slight significant confusion with pneumonia.

**Results**

The model achieved an accuracy of approximately 81.6% on the test set. Training and validation metrics indicate an overall satisfactory performance. The model demonstrates proficiency in recognizing patterns specific to each class. However, misclassifications are noted, particularly between normal and pneumonia classes, as evident from the confusion matrix.

**Conclusion**

This CNN model represents a promising step in automated diagnosis through chest X-rays. Further optimization, possibly through hyperparameter tuning, an expanded dataset, or more sophisticated architectures, could enhance its diagnostic capabilities.

The model seems to perform well for pneumonia detection but shows room for improvement in correctly classifying normal cases and tuberculosis. The model's tendency to confuse COVID-19 with pneumonia and normal cases likely indicates that data representation might need to be reviewed. Increase in dataset could potentially improve distinction across classes. Further analysis using precision-recall curves, ROC curves, or other advanced metrics could provide more insight into the model's classification behavior as well.