**Intermediate Image Classification Assignment: Classify Chest X-Ray Images for Pneumonia Detection**

**Objective:**

Classify chest X-ray images into two categories: Pneumonia or Normal using a Convolutional Neural Network (CNN).

**Dataset:**

Chest X-Ray Images (Pneumonia) Dataset

**Steps:**

1. **Data Loading and Exploration:**
   * Download the Chest X-Ray Images dataset from Kaggle.
   * Load the dataset using a library such as TensorFlow or PyTorch.
   * Perform initial data exploration to understand the dataset's structure and distribution.
   * Visualize a sample of images from each class.
2. **Data Preprocessing:**
   * Normalize the image pixel values to a range of 0 to 1.
   * Perform data augmentation (e.g., rotations, flips, shifts) to increase the diversity of the training data and reduce overfitting.
   * Resize the images to a consistent size suitable for your CNN model (e.g., 128x128 or 224x224).
3. **Model Building:**
   * Build a CNN model using a framework such as TensorFlow/Keras or PyTorch.
   * Start with a simple architecture and gradually increase its complexity.
   * Use layers such as Conv2D, MaxPooling2D, Flatten, Dense, Dropout, etc.
   * Experiment with different architectures (e.g., adding more convolutional layers, using Batch Normalization).
4. **Model Training:**
   * Compile the model with an appropriate loss function (e.g., binary cross-entropy) and optimizer (e.g., Adam).
   * Train the model on the training data and validate it on the validation data.
   * Use callbacks such as ModelCheckpoint and EarlyStopping to save the best model and prevent overfitting.
   * Plot the training and validation accuracy and loss over epochs to monitor the training process.
5. **Model Evaluation:**
   * Evaluate the final model on the test set using accuracy and other relevant metrics (e.g., precision, recall, F1 score, AUC-ROC).
   * Generate a confusion matrix to understand the model's performance for each class.
6. **Model Improvement:**
   * Perform hyperparameter tuning using techniques such as Grid Search or Random Search.
   * Experiment with advanced techniques like Transfer Learning by using a pre-trained model (e.g., VGG16, ResNet50) and fine-tuning it on the chest X-ray dataset.
   * Compare the performance of different models and architectures.
7. **Model Saving and Loading:**
   * Save the trained model to disk using the appropriate library functions.
   * Implement a function to load the saved model and use it for making predictions on new images.
8. **Documentation:**
   * Document each step of your process in a Jupyter Notebook.
   * Include explanations, visualizations, and code comments to make your work reproducible.
9. **Report:**
   * Write a summary report highlighting the key findings, model performance, and any challenges faced during the process.

**Deliverables:**

1. A Jupyter Notebook containing:
   * Data exploration and preprocessing steps.
   * Model building, training, and evaluation.
   * Hyperparameter tuning and model improvement.
   * Visualizations and plots.
2. The trained model saved to disk.
3. A summary report in markdown or PDF format.

**Optional Challenges:**

1. **Cross-validation:** Implement k-fold cross-validation to ensure the robustness of your model.
2. **Ensemble Methods:** Combine predictions from multiple models to improve accuracy.
3. **Advanced Architectures:** Experiment with more advanced architectures like DenseNet, Inception, or EfficientNet.
4. **Explainability:** Use techniques like Grad-CAM to visualize which parts of the image the model is focusing on.

**Getting Started:**

Here are some resources to help you get started:

* **Chest X-Ray Images (Pneumonia) Dataset:** Kaggle Dataset
* **Tutorials:**
  + TensorFlow/Keras: TensorFlow Image Classification Tutorial
  + PyTorch: PyTorch Transfer Learning Tutorial

Good luck with your assignment! This should help you solidify your understanding of CNNs and image classification in a medical context.