Assignment 05: Hyperparameter Tuning and Model Optimization

Objective:
Optimize a Convolutional Neural Network (CNN) model for **animal image classification** using **hyperparameter tuning** techniques.

Tasks:

1. **Data Preparation**:
   - Load the **animal image dataset**.
   - Resize all images to **128x128** and normalize the pixel values to `[0, 1]`.
   - Split the dataset into **train (70%)**, **validation (15%)**, and **test (15%)** sets.

---

2. **Build a Baseline CNN**:

   - Design a **baseline CNN** model with:
     - 2 convolutional layers with **ReLU activation**.
     - **MaxPooling** after each convolution.
     - A fully connected **dense layer** with 128 units.
     - A **softmax output layer** for classification.

3. **Hyperparameter Tuning**:

   - Use **Keras Tuner** or **Grid Search** to tune the following hyperparameters:
     - Number of **convolutional filters** (e.g., 32, 64, 128).
     - **Kernel size** (e.g., 3x3, 5x5).
     - Dropout rate (e.g., 0.2, 0.3, 0.5).
     - Batch size (e.g., 16, 32, 64).
     - Learning rate (e.g., 0.001, 0.0001).
   - Use **validation accuracy** as the evaluation metric.

---

4. **Train the Optimized Model**:
   - Train the best model found from hyperparameter tuning for **15 epochs**.
   - Use **EarlyStopping** to stop training if the validation loss stops improving.

---

5. **Model Evaluation**:
    - Evaluate the optimized model on the **test set**.
    - Report the following:
        - **Test accuracy**.
        - **Confusion Matrix**.
        - Precision, Recall, and F1-Score.

---

6. **Visualization**:
    - Plot:
        - **Training vs. Validation accuracy/loss** for the optimized model.
        - The **Confusion Matrix** for test predictions.

---

7. **Analysis**:
    - Compare the performance of the **baseline model** and the **optimized model**.
    - Write a brief report discussing:
        - Which hyperparameters improved performance the most.
        - Observations on training time and accuracy.

Bonus Task (Optional):

- Use **TensorBoard** to track and visualize the training progress, including loss and accuracy metrics.

Submission Requirements:

- Submit a **Jupyter Notebook** with:
    - Code for each step (data preparation, model building, hyperparameter tuning, evaluation).
    - Final performance metrics.
    - Visualizations and a short analysis.