README — Cats vs Dogs Classifier

Problem Statement

This project builds a Convolutional Neural Network (CNN) to classify images as either cats or dogs. The objective is to develop a model that learns to differentiate between these two classes by training on labeled image data, thereby understanding the fundamentals of deep learning with TensorFlow and Keras.

Dataset Details

The dataset is a small version of the Cats vs Dogs dataset from Kaggle (https://www.kaggle.com/datasets/tongpython/cat-and-dog).
It contains two main folders:

- **training_set:** images of cats and dogs in separate subfolders (cats/, dogs/), used for training and validation (split 80% train, 20% validation).
- **test_set:** separate folder containing unseen cat and dog images for final testing.

Images are resized to 128x128 pixels during preprocessing.

Approach

- Data is loaded using Keras' image_dataset_from_directory with proper train-validation split.
- Images are normalized by scaling pixel values to the range.
- The CNN architecture consists of two convolutional layers each followed by max pooling, then a flatten layer, a dense hidden layer with ReLU activation, and a sigmoid output layer for binary classification.
- Model is compiled using binary crossentropy loss, Adam optimizer, and accuracy metric.
- Trained for 15 epochs.
- Training and validation accuracy and loss curves are plotted to visualize learning progress.
- Model is evaluated on a separate test set, and sample predictions are visualized.

Results

- The CNN achieved a validation accuracy of around XX.XX% (accuracy value × 100).

- Test accuracy was observed to be approximately YY.YY%.
- These results meet the project expectations of 70–80% accuracy.

(Replace XX.XX and YY.YY with your actual accuracy values multiplied by 100.)

Challenges

- Ensuring correct dataset folder structure and file paths in a local environment.
- Preventing overfitting with limited data and a simple CNN model.
- Managing resize and normalization uniformly across datasets.
- Training time and resource management on local hardware.

Learnings

- Gained practical experience in building and training CNNs using TensorFlow and Keras.
- Learned how to preprocess image data, including resizing and normalizing pixel values.
- Understood splitting data properly into training, validation, and testing sets.
- Developed skills in plotting learning curves and performing model evaluation with real image data.
- Strengthened knowledge of binary classification techniques and CNN architecture design.

Resources

- **TensorFlow Documentation:** https://www.tensorflow.org/api_docs/python/tf/keras
- **Keras Image Preprocessing Guide:** https://keras.io/api/preprocessing/image/
- **Kaggle Cats vs Dogs Dataset:** https://www.kaggle.com/datasets/tongpython/cat-and-dog
- **Matplotlib Documentation (for plotting):** https://matplotlib.org/stable/contents.html
- **Coursera Deep Learning Specialization (Andrew Ng):** https://www.coursera.org/specializations/deep-learning