**Project Summary**

This project focuses on creating a **convolutional neural network (CNN)** designed to distinguish between images of cats and dogs. The process involves several key phases:

**1. Model Training:**

The CNN is trained using a dataset that comprises:

* 4000 dog images
* 4000 cat images

**2. Model Evaluation:**

Once the training phase is complete, the model’s performance is tested on a separate evaluation dataset, which contains:

* 1000 dog images
* 1000 cat images

**3. Prediction Deployment:**

After achieving sufficient accuracy, the trained CNN will be employed to classify fresh images stored in the single\_prediction folder into either the "dog" or "cat" category.

**Tools and Technologies**

This project utilizes **TensorFlow** along with its high-level API, **Keras**, to build and train the CNN. TensorFlow acts as the computational backbone, supporting both CPU and GPU operations, while Keras simplifies the process of designing and fine-tuning neural network architectures.

**Why Choose TensorFlow and Keras?**

* **TensorFlow**: A scalable, high-performance framework for building machine learning and deep learning models. It efficiently supports hardware acceleration for quicker computation.
* **Keras**: A user-friendly API built for simplicity, enabling easier creation and experimentation with neural networks. Its integration into TensorFlow provides seamless access to powerful features without compromising code simplicity.

**Step-by-Step Approach**

1. **Data Preparation:**
   * Arrange the dataset into well-structured directories for training and testing purposes.
   * Apply preprocessing techniques like **image normalization** (scaling pixel values) and **data augmentation** (random transformations) to enhance model accuracy and reduce overfitting.
2. **CNN Model Construction:**
   * Design the CNN using TensorFlow/Keras by stacking layers such as:
     + **Convolutional layers** for feature extraction
     + **Pooling layers** for dimensionality reduction
     + **Fully connected layers** for final classification
3. **Model Training:**
   * Train the network using the prepared training dataset.
   * Monitor progress through metrics such as loss and accuracy to evaluate performance.
4. **Model Evaluation:**
   * Evaluate the trained model on the test dataset to measure its generalization ability.
5. **Predictions on New Images:**
   * Utilize the model to classify images stored in the single\_prediction directory as either dogs or cats.

**Folder Organization**

The project follows this folder layout:

lua

Copy code

project/

│

│-- training\_dataset/

│ │-- dogs/

│ │-- cats/

│

│-- test\_dataset/

│ │-- dogs/

│ │-- cats/

│

│-- single\_prediction/

│ │-- <new images for prediction>

│

│-- ReadMe

│-- <additional project scripts and files>

**Getting Started**

**1. Install Required Dependencies:**

Before running the project, ensure you have TensorFlow installed:

bash

Copy code

pip install tensorflow

**2. Run the Training Script:**

Execute the provided script to train and evaluate the CNN model.

**3. Classify Your Images:**

Place any new images into the single\_prediction folder. Run the model inference to get predictions on these images.

**Notes to Keep in Mind:**

* Ensure the dataset is correctly structured and preprocessed before initiating model training.
* Input images in the single\_prediction directory must be in supported formats such as **JPEG** or **PNG**.
* If possible, use a machine equipped with a **GPU** to accelerate training time and computations.