Functions:

--load\_data  
Loads cat and dog images and organizes them into a usable structure.

--preprocess\_data  
Prepares the dataset by resizing images, normalizing pixel values, and possibly converting images into feature vectors. This ensures the data is ready for machine learning models.

--split\_and\_save\_dataset

1. Dataset splitting (Stratified Split):
   * First, an index array is created for all images (indices = np.arange(len(images))).
   * Then, using train\_test\_split, the dataset is divided into:
     + Training set
     + Test set
   * Because stratify=labels is used, the class balance (cats vs. dogs) is preserved in both sets. For example, if the dataset is 50% cats and 50% dogs, both train and test sets keep that same ratio.
2. Feature extraction (HOG Features):
   * The images for training (train\_images) and testing (test\_images) are selected based on their indices.
   * Then, the extract\_hog\_features function is applied to extract HOG descriptors from each image.
   * These extracted features (X\_train and X\_test) are what the machine learning model will actually use as input.
3. Directory creation
4. Saving images:
   * For each image (both training and test):
     + Based on its label (0 = cat, 1 = dog), the image is placed into the correct folder (cats or dogs).
     + The filename is generated as class\_index.png (e.g., cats\_12.png).
     + Since the images were normalized to values between 0 and 1, they are converted back to the 0–255 range and saved in PNG format.
5. Printing statistics:
   * At the end, the function prints useful dataset statistics:
     + Number of images in the training and test sets.
     + Number of cats and dogs in each set.
6. Function return values:
   * X\_train, X\_test: Extracted HOG features.
   * y\_train, y\_test: Corresponding labels (0 = cat, 1 = dog).
   * idx\_train, idx\_test: The indices of the images assigned to train and test sets.

--train\_random\_forest

This function trains a Random Forest classifier on HOG features.

Steps:

1. Inputs:
   * X\_train: HOG features of training images.
   * y\_train: Labels (0 = cat, 1 = dog).
   * n\_estimators: Number of trees (default 100).
   * random\_state: For reproducibility.
2. Model setup:  
   Initializes RandomForestClassifier with parameters:
   * max\_depth=20: limit tree depth.
   * min\_samples\_split=5: minimum samples to split a node.
   * min\_samples\_leaf=2: minimum samples per leaf.
   * n\_jobs=-1: use all CPU cores.
3. Training:  
   Fits the model on training data (clf.fit).
4. Output:  
   Returns the trained model clf, ready for prediction and evaluation.

--evaluate\_model  
The evaluate\_model function tests the trained Random Forest on unseen data by predicting labels for the test set (y\_pred) and then comparing them with the true labels to calculate accuracy. It returns both the accuracy score, which shows overall performance, and the predicted labels for further analysis.

Improvement:  
Replaced Decision Tree with Random Forest, which improved accuracy and model stability.

**HOG (Histogram of Oriented Gradients)**: یکی از روش‌های استخراج ویژگی در پردازش تصویر است که به طور گسترده در کاربردهای تشخیص اشیاء مورد استفاده قرار می‌گیرد. این روش بر اساس محاسبه گرادیان‌های محلی تصویر و توزیع جهت‌گیری آنها عمل می‌کند. ابتدا گرادیان‌های افقی و عمودی تصویر محاسبه شده و سپس بزرگی و جهت گرادیان در هر پیکسل تعیین می‌شود. تصویر به سلول‌های مربعی تقسیم شده و برای هر سلول، هیستوگرامی از جهت‌های گرادیان تشکیل می‌شود. این هیستوگرام‌ها پس از نرمال‌سازی، به یکدیگر متصل شده تا بردار ویژگی نهایی حاصل شود. HOG به دلیل مقاومت در برابر تغییرات نوری و هندسی کوچک، در تشخیص انسان عملکرد مطلوبی دارد.

**Decision Tree** :رخت تصمیم الگوریتمی در یادگیری ماشین نظارت شده است که داده‌ها را بر اساس ویژگی‌هایشان به صورت سلسله‌مراتبی یا طبقه‌بندی پیش‌بینی می‌کند. این ساختار شامل گره‌های داخلی (شرط ها یاهمان سوال هایی که میپرسیم )، شاخه‌ها (که نتایج سوال را نشان می‌دهند) و گره‌های برگ (که شامل تصمیم نهایی هستند) می‌باشد. فرآیند ساخت درخت بر اساس معیارهایی چون اطلاعات متقابل (Information Gain) یا ضریب جینی (Gini Index) انجام می‌شود. مزیت اصلی این روش قابلیت تفسیر بالا و سادگی درک آن است، اما معایبی مثلoverfitting دارد.

**Random Forest :**جنگل تصادفی یک روش یادگیری   
ترکیبی (Ensemble Learning) است که از مجموعه‌ای از درختان تصمیم برای بهبود عملکرد پیش‌بینی استفاده می‌کند. این الگوریتم بر اساس دو اصل Bootstrap Aggregating (Bagging) و انتخاب تصادفی زیرمجموعه‌ای از ویژگی‌ها عمل می‌کند. هر درخت روی نمونه‌ای تصادفی از داده‌های آموزشی (با جایگذاری) آموزش دیده و در هر گره تنها زیرمجموعه‌ای از ویژگی‌ها برای تقسیم‌بندی در نظر گرفته می‌شود. پیش‌بینی نهایی از طریق رای‌گیری اکثریت (در مسائل طبقه‌بندی) یا میانگین‌گیری (در مسائل رگرسیون) انجام می‌شود. این روش مزایایی همچون کاهش overfitting، مقاومت در برابر نویز و عملکرد مناسب روی انواع مختلف داده‌ها دارد.