Explanations_Task4

The Work Flow:

1)
The zip file is uploaded and extracted.

2)
Images are read, converted to grayscale, resized, and the lists images and labels are created.

3)
HOG features are computed for all images, and HOG images are also generated.

4)
The data is split into train/test sets in a stratified way; images are saved in separate folders; HOG features for train/test are extracted separately, and a Decision Tree model is trained on them.

5)
The model is evaluated on the test set, and a few samples are displayed showing the original image alongside its HOG image.

A summary of the explanation for each functio

1. FUNCTION: extract_hog_features(images, visualize=False)
PURPOSE: Extracts HOG (Histogram of Oriented Gradients) features from grayscale images. HOG captures edges, gradients, and shape information which is useful for classification.
HOW IT WORKS:

* Loops over each image in the list.
* For each image, calculates the gradient histogram in small cells and normalizes over blocks.
* If visualize=True, it also generates the HOG image for visual inspection.
* Appends the features (and HOG images if visualize=True) to lists.
* Converts lists to numpy arrays and returns them.
KEY POINTS:
* orientations, pixels_per_cell, and cells_per_block control the resolution of the descriptor.
* HOG features summarize structure, not raw pixels.

☆☆☆☆☆☆☆☆☆■■■■■☆☆☆☆☆☆☆

2. FUNCTION: split_and_save_dataset(images, labels, test_size=0.2, train_dir, test_dir)
PURPOSE: Splits the dataset into training and test sets, saves images to directories, and extracts HOG features for training and testing.
HOW IT WORKS:

* Uses train_test_split with stratify=labels to maintain class balance.
* Creates separate arrays for training and test images and labels.
* Creates train/test directories if they don’t exist.
* Saves each image as a PNG in the correct directory (train or test) with labels in the filename.
* Calls extract_hog_features on both training and test images to get HOG feature vectors.

* Returns HOG features, labels, and train/test indices.
KEY POINTS:
* Ensures balanced class distribution in splits.
* Prepares the dataset both as image files and as HOG feature vectors for model training.

☆☆☆☆☆☆☆☆☆■■■■■☆☆☆☆☆☆☆

3. FUNCTION: train_decision_tree(X_train, y_train)
PURPOSE: Trains a Decision Tree classifier on the training HOG features.
HOW IT WORKS:

* Initializes a DecisionTreeClassifier.
* Fits the classifier on the training features (X_train) and labels (y_train).
* Returns the trained classifier.
KEY POINTS:
* Decision Tree creates rules based on feature values to separate classes.

☆☆☆☆☆☆☆☆☆■■■■■☆☆☆☆☆☆☆

4. FUNCTION: evaluate_model(clf, X_test, y_test)
PURPOSE: Evaluates the trained classifier on the test dataset.
HOW IT WORKS:

* Uses the classifier to predict labels for X_test.
* Compares predicted labels with y_test to compute accuracy.
* Returns the accuracy and predicted labels.
KEY POINTS:
* Accuracy = (number of correct predictions) / (total number of samples).
* Measures how well the model generalizes to unseen data.

☆☆☆☆☆☆☆☆☆■■■■■☆☆☆☆☆☆☆

SUMMARY OF THE PIPELINE:

1. extract_hog_features → converts images to HOG feature vectors.
2. split_and_save_dataset → splits the dataset, saves images, and extracts HOG features for train/test.
3. train_decision_tree → trains the classifier on training HOG features.
4. evaluate_model → tests the classifier and reports performance metrics.