Assignment 3

Problem Statement:

"Implement a basic bag of words model to classify images (possibly perturbed) into one of 30 categories by training and validation on the dataset we provided, which contains 30 categories."

Software Used:

MATLAB R2018b

Solution:

In this assignment we aim to classify images in to one of 30 categories. There are two main parts of the solution:

1) To find the features of a given image set, we use a bag of visual words model. A bag of words is a representation of commonly occurring features in text or images. In this case, we use this model to represent visual features that occur in images.
2) To actually classify the images into categories, we use the K Nearest Neighbors (KNN) model. In this model, an image is assigned the category of the K images that are most similar to it.

The code is split into three parts: a feature extraction method, a method to perform the classification using KNN and a test script used to evaluate the solution.

In the feature extraction method, we begin by performing pre-processing actions on the image. The image is converted into grayscale using the **rgb2gray()** function. Next, the image is filtered using a Gaussian filter to remove noise and reduce the possibility of detecting false features. This is done using **imgaussfilt2()**. Finally, we use the function **extractLBPFeatures()** to find the features of the image. This function extracts features in the form of Local Binary Patterns (LBPs). LBPs are visual descriptors which label pixels by thresholding their neighboring pixels. The descriptors are binary in nature. The function returns a vector containing all the features of an image.

In the KNN method, we assign labels to specific sets of training data based on the features found during feature extraction. We then create a KNN classification model on the testing images using the labels assigned to the training data. This is done using **fitcknn()**. Finally, we predict the value of the test set using this model using the **predict()** function. This function returns the labels of the testing data.

The testing script is used to create the test set, call the feature extraction method to acquire visual features, to assign the actual labels to the testing data and check the accuracy of the solution.

Algorithm

1) Feature Extraction
    a. Obtain image from calling function
    b. Convert image to grayscale
    c. Remove noise by filtering
    d. Extract LBP Features
    e. Return feature vector

2) K Nearest Neighbor
    a. Obtain feature vector of the testing data
    b. Create a feature vector of the training data
    c. Create a KNN model based on the training data
    d. Use the KNN model to predict the labels of the testing data
    e. Return the labels of the testing data

Results

This algorithm yields an accuracy of 41.67%. Alterations to the values in the feature extraction and KNN functions make minor changes to the value of accuracy.