Image Classification using CIFAR-10 Dataset: Applying K-Nearest Neighbors Classifier using Euclidean (L2) Distance with 5-fold Cross-Validation

In this assignment, we will explore the CIFAR-10 dataset, which consists of 60,000 32x32x3 color images in 10 different classes. The objective is to build a model for image classification by applying Euclidean (L2) distances and using 5-fold cross-validation to find an optimal value of k. The training data consists of 50,000 images, and testing data consists of 10,000 images. We will use a subset of the training and testing data to perform classfication due to memory limitations. We will plot a graph to visualize the accuracy of different hyperparameter values (K) and obtain accuracy on testing data using the optimal k value. for this particular dataset. You will write your code for this assignment in the ‘KNN\_assignment.py’ in the parts of the script where you are instructed to implement code. You will also provide a report (in pdf or word doc format) of the results obtained in the assignment.

**Dataset:**

The CIFAR-10 dataset can be downloaded from the following link: <https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz>. Code is provided to download and extract the dataset in your current directory in the following path: *./data/cifar-10-batches.py*. Run the code to obtain the dataset or manually obtain it in the path given. Code provided will also extract training and testing data from the dataset.

**Model Building and Evaluation:**

To classify the gray-scale CIFAR-10 images, we will employ the k-nearest neighbors (k-NN) algorithm. This algorithm classifies an image by considering the class labels of its k nearest neighbors. We will use the Euclidean (L2) distance metric, to measure the similarity between images. You will need to implement code to apply the following and include comments to explain your code:

Preprocess the data and implement methods to compute Euclidean distances and predict labels of testing data. You will obtain model accuracy on test data using k=5. Then, you will evaluate the performance of model on a set of k values using 5-fold cross-validation. The training dataset will be divided into 5 subsets (folds), and for each value of k, the model will be trained and evaluated 5 times using different combinations of training and validation data. This approach allows us to assess the model's performance across multiple folds, providing a more robust evaluation. You will then use the best value of k from your evaluation to obtain model accuracy on testing data.

**Results and Analysis:**

For each fold of cross validation, you will record the accuracy achieved by the model for each value of K. You will compute the average accuracy across all folds for each value of K and plot a visualization of the accuracies obtained using cross validation.

Provide all results from running the model in your report, including those mentioned above, specifying what the results are. In your report, Discuss why you performed cross validation to find k, including how an optimal value of k led to better fitting of the model. Also discuss how the performance of this classifier would compare to a more sophisticated model, such as a Neural Network (NN) or Convolutional-NN on the CIFAR-10 dataset.

**Bonus:**

To obtain bonus points, obtain the results mentioned above using Manhattan (L1) distance as the distance metric of your classifier and compare model performance using Manhattan distance to that using Euclidean distance.

**Submission:**

Modify only the ‘KNN\_assignment.py’ script in the assignment folder and upload your completed code, results and report without the downloaded data (this would take too much space) in the same folder in zip format to the provided submission link. My solutions required 27 lines of code.