**Readme**

**Files Used:-**

1. pcaq1.py
2. hogq1.py
3. q2.py

There are three questions in the assignment, and the first two python files contain a solution to question 1, while the third python file contains solutions to questions 2 and 3.

**About pcaq1.py :**

1. This python file contains the functions that perform multiclass classification on the CIFAR dataset with PCA as a feature descriptor.
2. The functions with the functionalities in this file are given below :

* **accuracy(self, actual, predicted):** Calculates the accuracy using actual class labels and predicted class labels.
* **Unpickle (file):** Loads the training and testing data from the CIFAR dataset batch file. Pickel module is used for the functionality.
* **UPCA(Data,n):** Reduces the dimensionality of the data ‘Data’ using PCA such that the new data retains the 90% variance of the original data.
* **UTSNE(Data, Label) :**
  + Reduces the dimensionality of the data ‘Data’ using TSNE with the number of components
  + Visualize the new dataset with 2 dimensions.
* **GridCV(XTrain, YTrain):**
* Performs the grid search on the parameter grid with k = 5.
* Saves the grid search object model.
* **GetOptimalValues(gridcv,XTrain,YTrain,XTest,YTest) :** 
  + Use the above-saved grid search object to extract the optimal parameter values.
  + Fits the new SVM model with the above grid value and saves it.
  + Calculates the training and testing accuracy.
* **NONSV(XTrain,YTrain,XTest,YTest):** 
  + Use the above-saved SVM model to extract the support vectors which are used as new Training Data
  + Use the above-saved grid search object to extract the optimal parameter values.
  + Fits the new SVM model with the above grid value and new training data
  + Saves the model.
  + Calculates the training and testing accuracy.

**About hogq1.py :**

1. This python file contains the functions that perform multiclass classification on the CIFAR dataset with HOG + Color Histogram as a feature descriptor.
2. The functions with the functionalities in this file are given below :

* **accuracy(self, actual, predicted):** Calculates the accuracy using actual class labels and predicted class labels.
* **Unpickle (file):** Loads the training and testing data from the CIFAR dataset batch file. Pickel module is used for the functionality.
* **HOG(gray\_img):** Extracts the HOG features from the set of grayscale images
* **Hist(hue\_img):** This function is used to find the color histogram of the image.
* **UTSNE(Data, Label) :**
  + Reduces the dimensionality of the data ‘Data’ using TSNE with the number of components
  + Visualize the new dataset with 2 dimensions.
* **GridCV(XTrain, YTrain):**
* Performs the grid search on the parameter grid with k = 5.
* Saves the grid search object model.
* **GetOptimalValues(gridcv,XTrain,YTrain,XTest,YTest) :** 
  + Use the above-saved grid search object to extract the optimal parameter values.
  + Fits the new SVM model with the above grid value and saves it.
  + Calculates the training and testing accuracy.
* **NONSV(XTrain,YTrain,XTest,YTest):** 
  + Use the above-saved SVM model to extract the support vectors which are used as new Training Data
  + Use the above-saved grid search object to extract the optimal parameter values.
  + Fits the new SVM model with the above grid value and new training data
  + Saves the model.
  + Calculates the training and testing accuracy.

**About q2:**

1. q2.py contains the class named SVM class.
2. SVM contains the following member functions with the functionalities :

* **Fit(self, XTrain, YTrain):** Fits the SVM model either with linear or RBF kernel based on the requirement.
* **Predict(self, XTest):** Predicts the class labels of the data points X\_test.

3. The q2.py contains the following utility functions :

* **accuracy (actual, predicted):** Calculates the accuracy using actual class labels and predicted class labels.
* **TrainTestSplit(Data,r):** Splits the dataset into training and testing data with a test size ratio of r.
* **Classwise\_Accuracy(Ypred, Yact):** Calculates each class's classwise accuracy using actual class labels and predicted class labels.
* **NfoldSplit(Data,n):** Splits the Data into n folds.
* **loaddata(path):** Loads the dataset from the specified path.
* **plot(df):** Creates the scatter plot of the data frame df.
* **plotdecisionbounday(X,Y,s,sup,kernel,C,g=None) :** 
  1. Plot the support vectors
  2. Plot the hyperplane
  3. Plot the margins.

**Steps to Run the code :**

1. Unzip the submitted files.
2. Store the files in the google drive.
3. To run the pcaq1.py :
   * 1. Store the data set in the drive
     2. Open the pcaq1.py in the google colab.
     3. Mount the drive.
     4. Set the right path argument, which contains the dataset, saved models, and data.
     5. Compile all the cells to get the required output.
4. To run the hogq1.py :
   * 1. Store the data set in the drive
     2. Open the hogq1.py in the google colab.
     3. Mount the drive.
     4. Set the right path argument, which contains the dataset, saved models, and data.
     5. Compile all the cells to get the required output.
5. To run the q2.py :
   * 1. Store the data set in the drive
     2. Open the q2.py in the google colab.
     3. Mount the drive.
     4. Set the right path argument, which contains the dataset, saved models, and data.
     5. Compile all the cells to get the required output.