**Ideas and Solutions**

Step 1: Summarize the known information

Save the information in the training data file in a dictionary to make it easier to connect the digit and the known forms.

Step 2: Develop the kNN model

Model 1: use the distance between two vectors to find the k nearest neighbors of an unknown written digit. Among the k neighbors, the final prediction will be the most frequent prediction

Model 2: for each digit, reduce all the vectors to one which only has the common part of all forms. k = 1. The final prediction will be the nearest neighbor of an unknown written digit.

Step 3: Compare the accuracy of different kNN models

Compare the accuracy of models with different k and prediction rules. And decide to use the one with the largest accuracy.

**Program Flow, Data Structures & Algorithm**

|  |  |  |  |
| --- | --- | --- | --- |
| funcitons | load\_data() | show\_training\_info() | show\_testing\_info() |
| input | the file name that will be used to formulate a kNN model | N/A | Two dictionaries containing the number of the correct predictions and the number of incorrect predictions for each specific digit |
| output | a dictionary including the digit and all the vectors of known handwritten forms | the training file information including the number of known handwritten forms for each number. | the accuracy of the kNN model |
| functionality | summarize the training data file into a dictionary | present the training data file information: the known statistics used to formulate the kNN model | present the result of the training-testing cycle |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| function | knn\_by\_majority() | data\_by\_or() | predict\_by\_knn\_dist() | predict\_by\_knn\_or() |
| input | the list of k nearest neighbors, sorted with the frequency of them | N/A | a vector of handwritten digit that will be predicted; k | a vector of handwritten digit that will be predicted; k |
| output | the prediction of the handwritten number | N/A | the predicted value | the predicted value |
| functionality | find out the number that occurs most often among the k nearest neighbors | reduce the vectors of each digit to one vector which contains the common area of them. The original training data set is changed into a new one where each of digit only has one known vector form | to predict an unknown handwritten form by finding out the digit of those vectors within short distance | to predict an unknown handwritten form by finding out optimal one among the nearest k neighbors |

|  |  |  |  |
| --- | --- | --- | --- |
| functions | compute\_accuracy() | predict() | main\_knn() |
| input | the name of the testing file; the knn model used to predict (predict\_by\_knn\_dist() or predict\_by\_knn\_or()) | the model used to predict (predict\_by\_knn\_dist() or predict\_by\_knn\_or()); the file that will be predicted | the model used to predict (predict\_by\_knn\_dist() or predict\_by\_knn\_or()) |
| output | The correct and the incorrect prediction; the accuracy of the kNN model | the predictions | The training info; the testing info. |
| functionality | To compare different kNN models and choose the optimal one with the highest accuracy | predict all the digits with the handwritten form in the predict data file | combine all the functions and structure the whole process |

**Limitations**

1. Although the prediction method of distance has very high accuracy, the error rate is still quite high when the handwritten digits 4 and 8 are the predicted result.
2. In the model using prediction method of distance, 7 may not be the optimal k. I found the k by estimating the general trend of the accuracy along with changing k manually and finally get the “best” one. A better k perhaps exists out of my reach.

**Enhancements**

1. For the first limitation, the accuracy of the kNN model can be further improved by specifically decreasing the error rate for the prediction of 4 and 8 with more kNN models. The programmer can randomly select several samples from the training data file to develop new kNN models to reexamine the handwritten form of digits when the existing method shows the prediction of 4 or 8. Only when the kNN models can intensify this prediction will their prediction be admitted. A further step could be selecting the most commonly predicted one among all the outcomes of different kNN models to be the final prediction.
2. For the second limitation, another function can be added to test the k in a wide range, for example, from 1 to 100, and then use a tuple to record the accuracy under each circumstance. A general trend will be presented clearly, so that the optimal k can be found. However, this method will be very time consuming. If the developer intends to randomly extract a sample from the training file to establish the kNN model instead of using the whole given file, the function needs to be called all the time to find a new optimal k.