1.0 Algorithm Selection

KNN: K-Nearest Neighbour classification algorithm is a relatively simple classification algorithm in machine learning. The algorithm method is simple, easy to understand, easy to implement, no need to estimate parameters, no training. Therefore, this experiment will use Python alone to implement the KNN algorithm. For the hyperparameters of the algorithm- ***Choose K value:*** For the choice of k value, If the value of k is small, the model will become complicated and prone to overfitting problems. But If the value of k is large, the prediction is prone to errors, and the model will become simple, and it does not have good generalization ability.

SVM: In this experiment, a nonlinear SVM was used. Since SVM has a better effect on classification problems, for non-linear problems only the inner product between the instance and the instance is involved, so there is no need to explicitly specify the non-linear transformation, but the inner product is replaced by the kernel function. Therefore, the author chose this algorithm and implemented it in the scikit-learn library. For the hyperparameters of the algorithm- ***The value of gamma:*** How to select the gamma value in the Gaussian kernel function is very important. If the gamma value is too small, underfitting will occur, and if the gamma value is too large, overfitting will occur. These conditions will directly affect the accuracy of the prediction results

Decision Tree: Since the decision tree can handle very complex problems, it can also fit complex data well. Therefore, in this experiment, the author chose the decision number based on the C4.5 algorithm as the classifier. For the selection of hyperparameters of the algorithm, the author pays more attention to the depth of the decision tree. ***Post-pruning:*** Decision tree is a non-parametric model. In order to prevent the occurrence of over-fitting, it is necessary to limit the degree of freedom of the decision tree by pruning.

2.0 Methodology

Data pre-processing and division:

After reading the image data of the training set and the image data of the test machine, normalize them at the same time. Since the data read is all RGB numbers, its value range is [0,255]. That is, all picture data is divided by 255 at the same time to achieve data normalization. Secondly, because the original data set does not contain the data of the validation set. Therefore, the author divided the 60,000 training set data into 10 evenly and took two of them as the verification set. Thus, the new data length is: ***Training set data [48000], Verifier data [12000], Test machine data [10000]***

Cross validation to obtain optimal hyperparameters and training model:

Randomly extract 1000 verification data from the verification set to select the best hyperparameters of each algorithm. ***KNN:*** Set the set of K values to [1,20] and K is a positive integer, and use the same verification data to obtain the accuracy rate returned by the corresponding K values. After obtaining all the maximum values of the accuracy of the returned results, the average of its K values is obtained as the optimal hyperparameter of the KNN algorithm. ***SVM:*** Similarly, the KNN algorithm, different gamma values are selected to get the accuracy of the returned result. Calculate the average of all gamma values corresponding to the maximum accuracy rate and use it as the optimal hyperparameter of the SVM algorithm. ***Decision Tree:*** Different from the previous two algorithms, the way to regularize the hyperparameters in the decision tree is to control the depth of the decision tree. In sk-learn, it is controlled by the hyperparameter max_depth. Reducing the value of max_depth will regularize the model and reduce the risk of overfitting.

Test and performance:

Before starting the test, the author divided the data in the test set again. Similar to the training set, the test set is divided equally into 10 parts. Each piece has 1000 data, and then 500 pieces of data are randomly selected from each piece of data. That is, each model is tested 10 times, and each test 500 data. Then take the average of the result set of each model as the final accuracy of each model. In addition, for the performance indicators of each model algorithm, the main concern is the running time of the algorithm and the accuracy of the algorithm.

3.0 Results

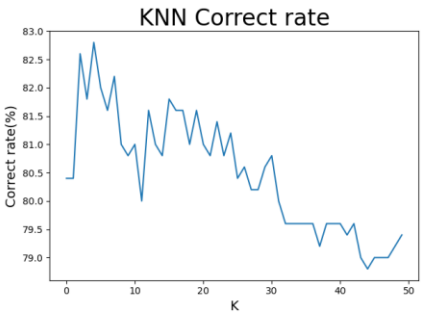

Figure 1

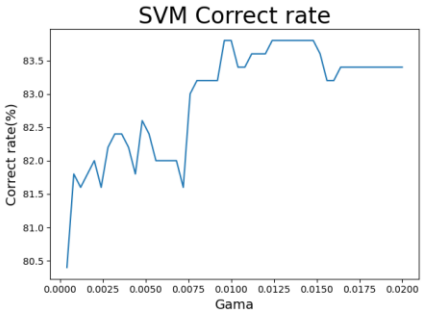

Figure 2

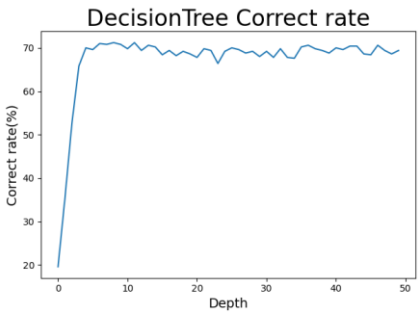

Figure 3

Algorithm	Hyperparameter	Time	Validation set correct rate	Test set correct rate
KNN	K = 4	2886.71s	80.63%	85.60
SVM	Gama = 0.0006	518.41s	82.54%	86.36%
Decision Tree	Depth = 9	56.96s	67.33%	79.82%

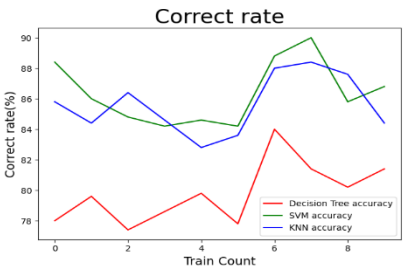

Figure 4

For Decision tree, it takes the least time, but the accuracy rate is the lowest among the three models. However, SVM takes moderate time and has the highest accuracy. Although the correct rate of KNN is like that of SVM, it takes the longest time.

4.0 Conclusion

The results of all algorithms are in line with experimental expectations, and the test results of all models trained in the experiment are better than the training results.

KNN: For the KNN algorithm, its accuracy is almost only related to the value of K. According to the data in Figure 1, the overall accuracy of the KNN algorithm decreases as the K value increases. When K takes a value of 5, the correct rate of the entire model is maximized. When the value of K is greater than 5, the accuracy of the model is significantly reduced. The classification boundary is blurred, which leads to the occurrence of underfitting events.

SVM: For the SVM algorithm, there are many hyperparameters that will affect the accuracy of the algorithm, such as different kernel functions and gamma values. In this experiment, we mainly study the influence of the gamma value on the accuracy of the algorithm, and the kernel function agrees to choose the Gaussian kernel function. The data in Figure 2 shows that when the gamma value is small or large, the accuracy of the algorithm is lower than that when the gamma value is moderate. When the gamma value is greater than 0.0075, the algorithm is prone to overfitting. As a result, the generalization ability of the model has declined.

Decision tree: For decision trees, this experiment only studies the depth of decision trees. The data in Figure 3 shows that when the depth of the decision tree is 9, its accuracy is the highest. The accuracy of the algorithm will decrease slightly if the depth is greater than 9, but the decrease is not large. In actual experiments, when the depth of the decision tree is greater than 9, the model will overfit. The test data cannot be classified well.

Optimal algorithm: For the training data, the author believes that SVM is the current best classification algorithm. Compared with the decision tree in terms of time consumed, although the training time of the SVM is relatively long reaching 518 seconds, the decision tree only needs 57 seconds. But in terms of accuracy, the accuracy of SVM is nearly 10% higher than that of decision trees. Compared with KNN, although the accuracy of KNN is like that of SVM. But in terms of time consumption, KNN is much larger than SVM. In addition, the data in Figure 4 shows that, compared with the other two algorithms, the fluctuation of the accuracy of support vector machine training is relatively flat, basically maintaining at 86%. Through the above three points of judgment, the author believes that SVM is the optimal algorithm compared to the other two algorithms on this training set.