# Assignment4 Report

1. **Data**

**1.1 Preprocessing**

First of all, the size of two datasets and each image is obtained, then, each file is tranformed into a matrix with size of [i, j, k], where ‘i’ refers to the number of samples, ‘j’ to the row number and ‘k’ to the column number. With this data format, we can easily extract the feature map using the average pooling method of each image. Last, “#” and “+” in images (Fig. 1) are replaced by “1”, empty values are replaced by “0”.

**1.2 Feature extraction**

As one digit image have small image size, every pixel of one image is regarded as a feature. However, face image has relatively large size, which will reduce the training speed and model accuracy(especially in NaiveBayes), so, the sum pooling method is also used for face images in the implementation of three algorithms. Comparison between single pixel feature and sum pooling is made.

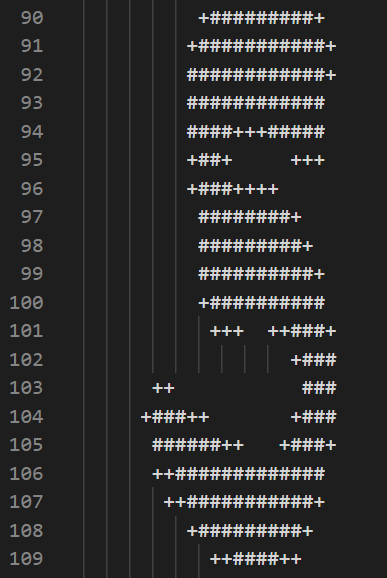


Fig. 1 One digit image sample

1. **Methods**

Face image detection corresponds to binary classification, and digit detection corresponds to multi-class classification.

* 1. **Perceptron**

The baisc idea of perceptron is to find a hyperplane(Eq. (1)), which will divide the input samples into two categories : positive () and negative ( = -1).

(1)

In the training process, and are firstly initialized with random values and 1 respectively. Then, if the output differs from the true label , and will be updated (Eq. (3) and (4)) by their differential (Eq. (5) and (6)) to the loss function (Eq.(2)):

(2)

(3)

(4)

(5)

(6)

Updation will stop if every is the same as .

Thee final and will be used to predict test samples and caculate model accuracy.

P erceptron’s linear hyperplane can only classify binary categories. In the implementation, perceptron method is redesigned by using max function, which introduces nonlinearity, to determine the label, thus, digit multi-class classification problem is successfully resolved.

Besides, we use pixel values (feature) as .

* 1. **Naive Bayes**

Naïve Bayes model is based on Bayes formula.

(7)

(8)

(9)

(10)

In the training process Eq. (7) is caculated by Eq. (8) and (9). Test data’s label probability is predicted by caculate P(label =i|test\_data\_feature) (Eq. (7)), and the final label is determined by the max label probability.

It must be stated that when feature(j) and certain label(i) do not appear at the same time in the training data, = 0, in order to avoid such situation, Laplace correction must be applied. In the code, is set to 0.001 when it’s value smaller than 0.001.

* 1. **Nerual Network**

The designed nerual network contains one input layer, one hidden layer and one output layer.

From input layer to hidden layer:

(11)

at hidden layer:

(12)

from hidden layer to output:

(13)

and is optimized by their differential to loss function (back propagation), loss function is defined by cross-entropy.

(14)

(15)

(16)

(17)

(18)

The optimization function will be processed for N iterations before final and come out.

Test data label is predicted by Eq. (13). For digit multi-class classification, in order to adapt to loss function(14), labels are transformed to one-hot format.

When compared perceptron with nerual network, perceptron performs like a simple one-layer network, but it can only work with binary classification.

1. **Results**

After times of implementaions, it is found that the best sum pooling size is 3\*3 for face image.

* 1. **Perceptron**

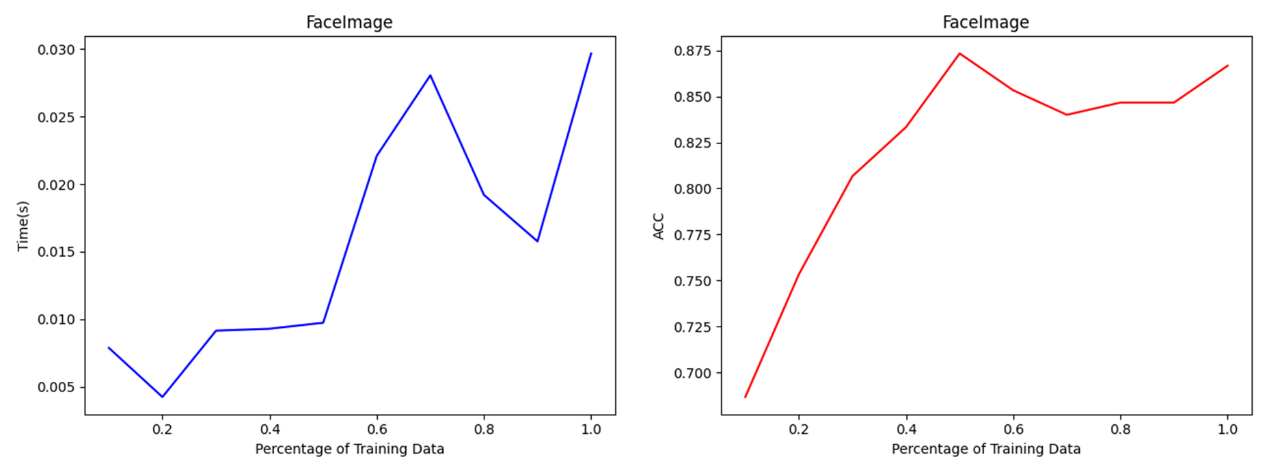


Fig. 2 the changing trend of time cost and accuracy with percentage of face image training data using perceptron method.

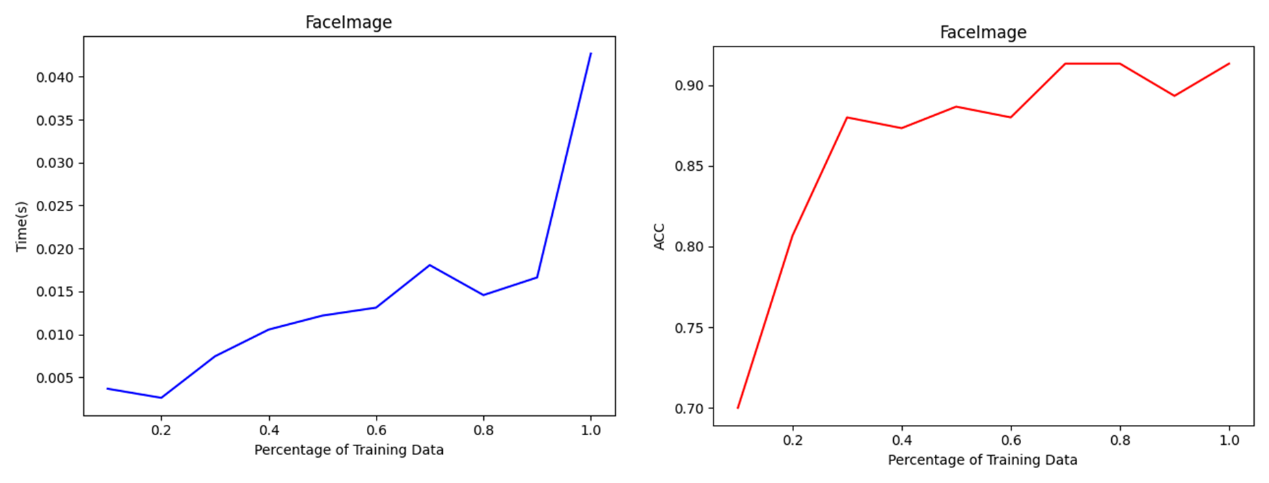


Fig. 3 same as Fig. 2, but with pooling size of 3\*3.

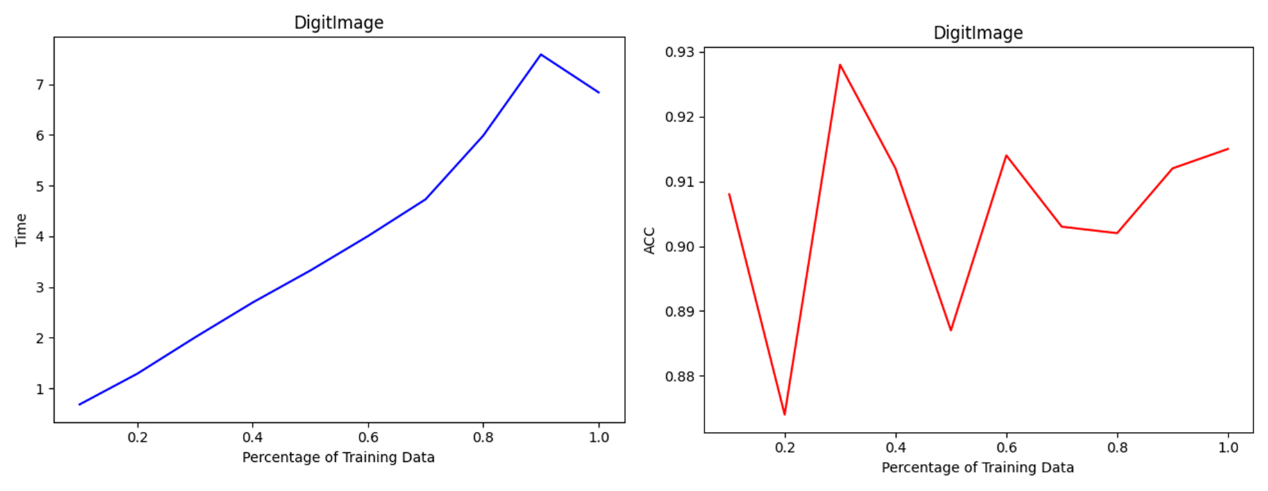


Fig. 4 same as Fig.2, but of digit training data

* 1. **NaiveBayes**

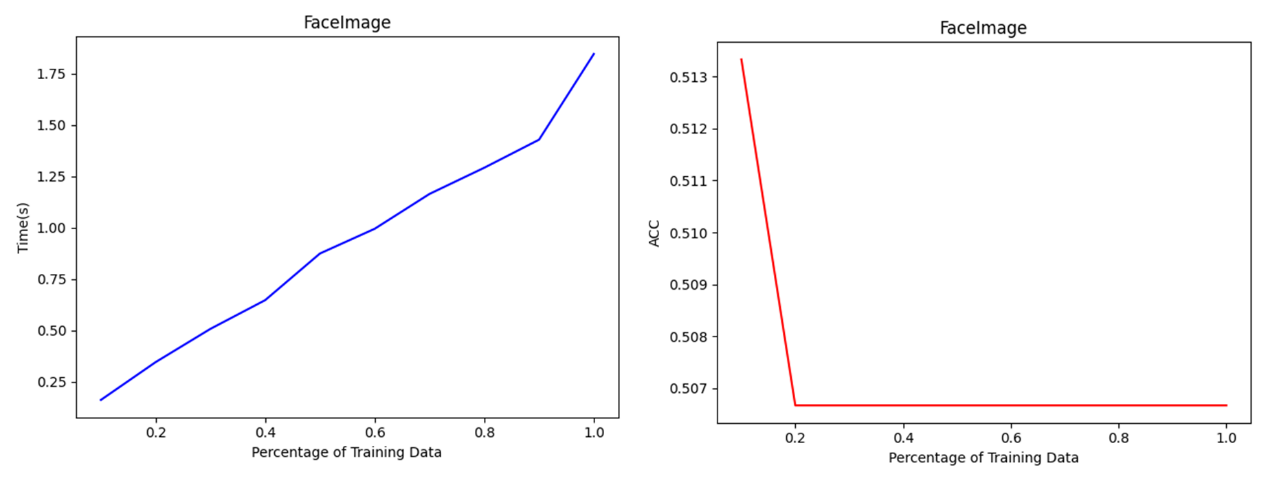


Fig. 5 the changing trend of time cost and accuracy with percentage of face image training data using naiveBayes method.

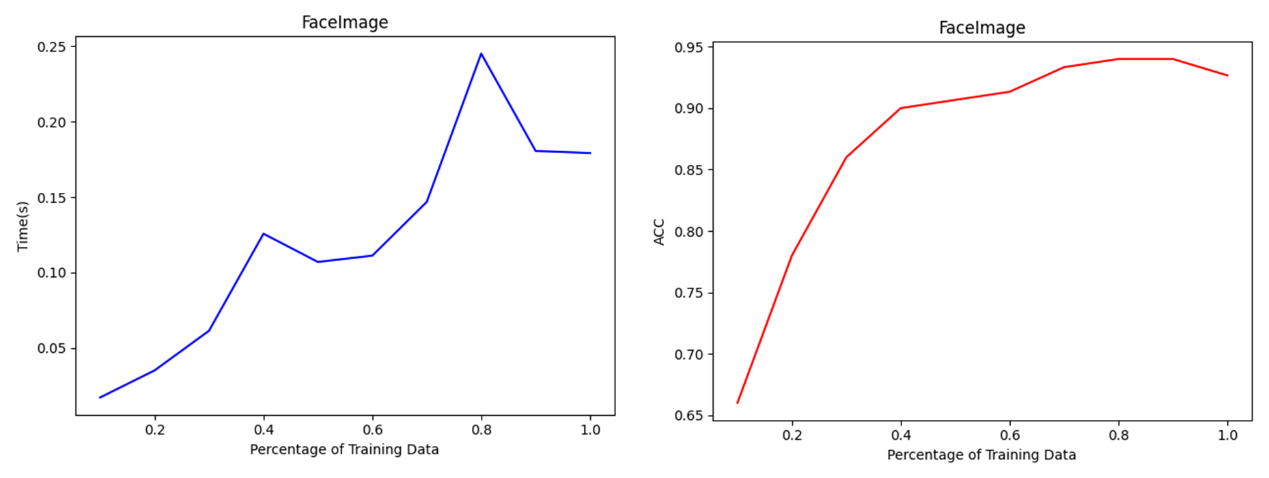


Fig. 6 same as Fig. 5, but with pooling size of 3\*3

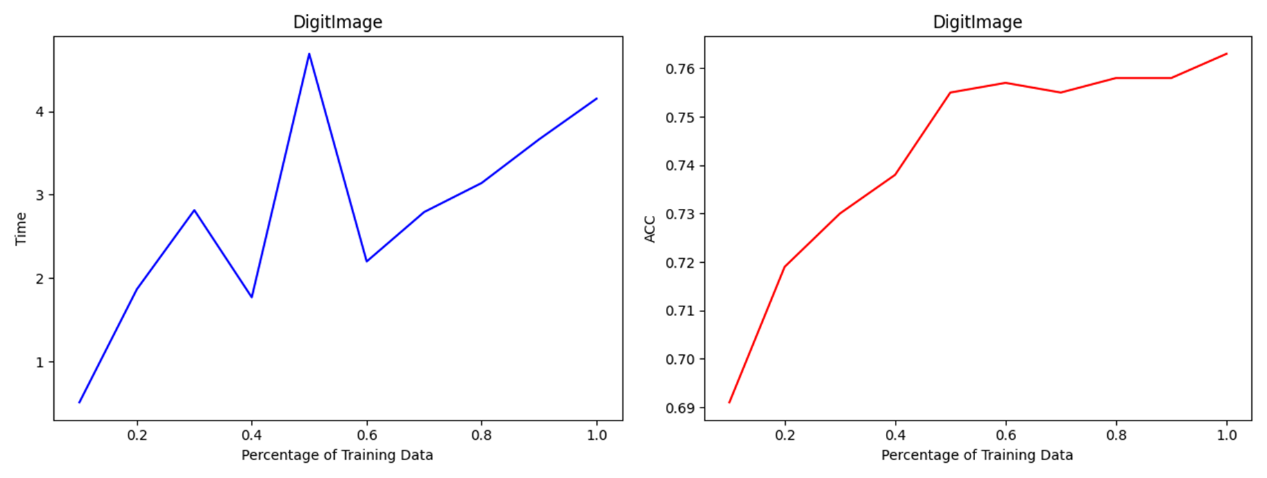


Fig. 7 same as fig. 5, but of digit image training data

**3.3 Neural Network**

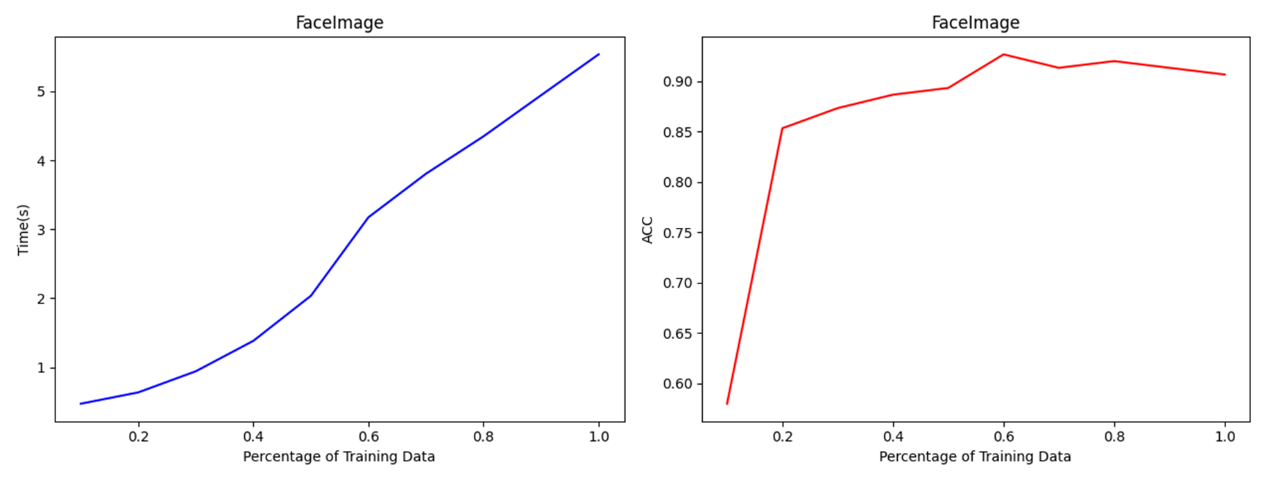


Fig. 8 the changing trend of time cost and accuracy with percentage of face image training data using neuralNetwork method.

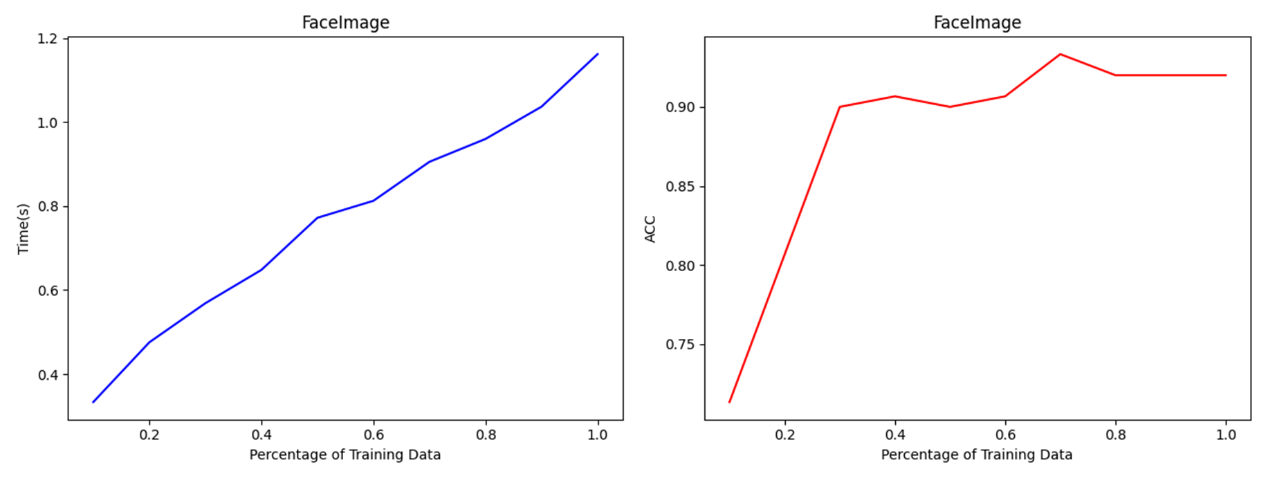


Fig. 9 same as fig. 8, but with pooling size of 3\*3.

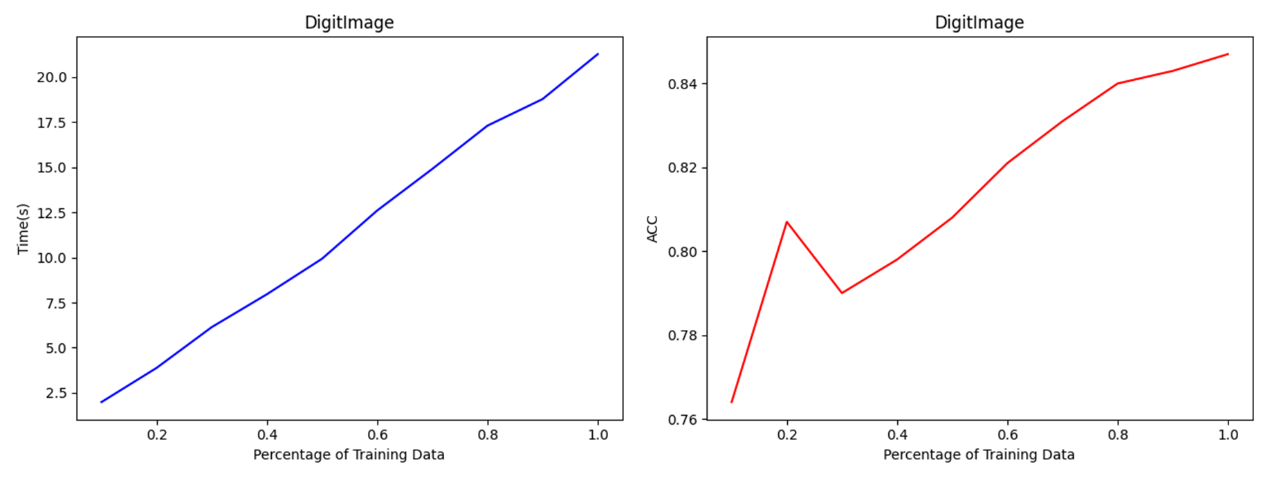


Fig. 10 same as fig.8, but of digit image training data.

* 1. **Accuracy**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Perceptron | Perceptron\_pool | NaiveBayes | NaiveBayes\_pool | NeuralNetwork | NeuralNetwork\_pool |
| Face image | 0.87 | 0.913 | 0.503 | 0.931 | 0.922 | 0.935 |
| Digit image | 0.925 |  | 0.864 |  | 0.847 |  |

1. **Conclusion**
   1. **Model performance on face images**

(1)It is clear that the training time of models are significantly reduced when using sum pooling method, meanwhile, higher classification accuracies appear.

(2)Pooling is not used in Fig. 5, and the performance of naïveBayes appears abnormal. The reason is that there are too many features of one image, and every feature’s label probability is small, which results in lots of near-zero value when calculate Eq. (7).

**4.2 Model performance on digit images**

(1)Without using pooling method, the training time of models are clearly high.

(2)Perceptron method performs best of three models.