Report TD ROB311 :
Facial Expression Recognition-KNN

Yufei HU

10/10/2021

Table des matières

1 Introduction

The subject of this experiment is to classify the emotion given a static face image with KNN. The dataset contains 28709 48*48 pixel grayscale images of faces which are divided into 6 different categories((0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). We will use a classifier to achieve this task which contains feature extraction technology and KNN supervised learning.

2 Our Work

2.1 Data loading process

Firstly, we unzip the facial database and put the train and test set into a folder named **'database'**. In order to check if we successfully load the images, we use the package of matplotlib to show 7 different emotions. The result is shown in Figure 1 as following.

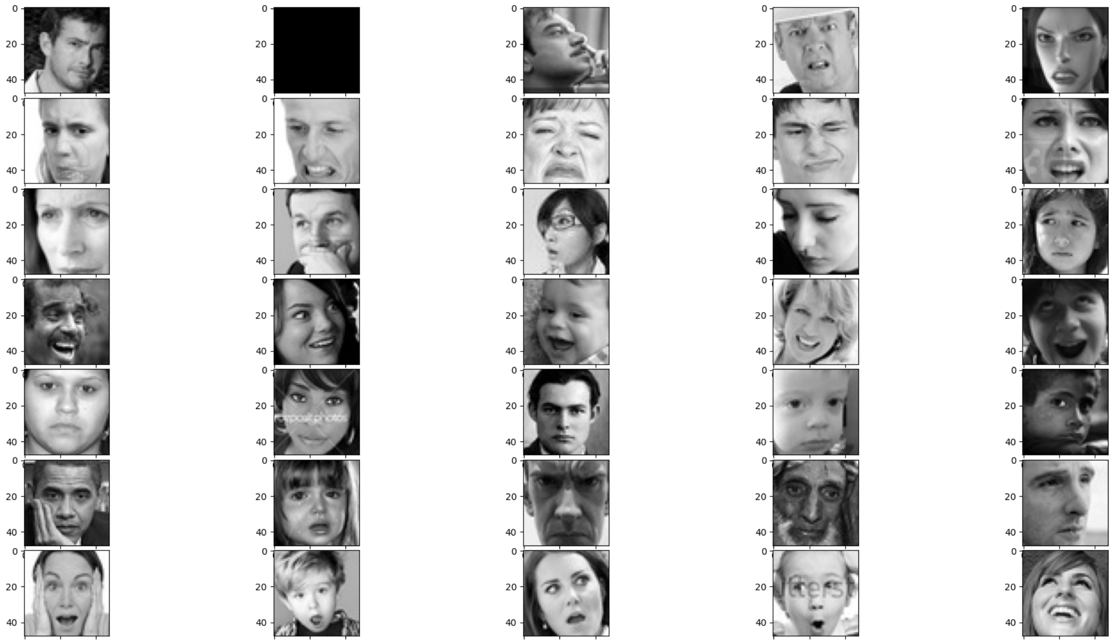


FIGURE 1 – The example of 7 different categories of facial emotions.

2.2 Feature extraction with LBP

LBP (Local Binary Pattern) is an operator used to describe the local texture features of an image. It has significant advantages such as rotation invariance and gray level invariance, and is used to extract local texture features of an image.

In our experiment, we will use LBP algorithm from the scikit-learn package as the feature extractor. When we apply the LBP to the grayscale images in the database, we can get an example shown as Figure 2.

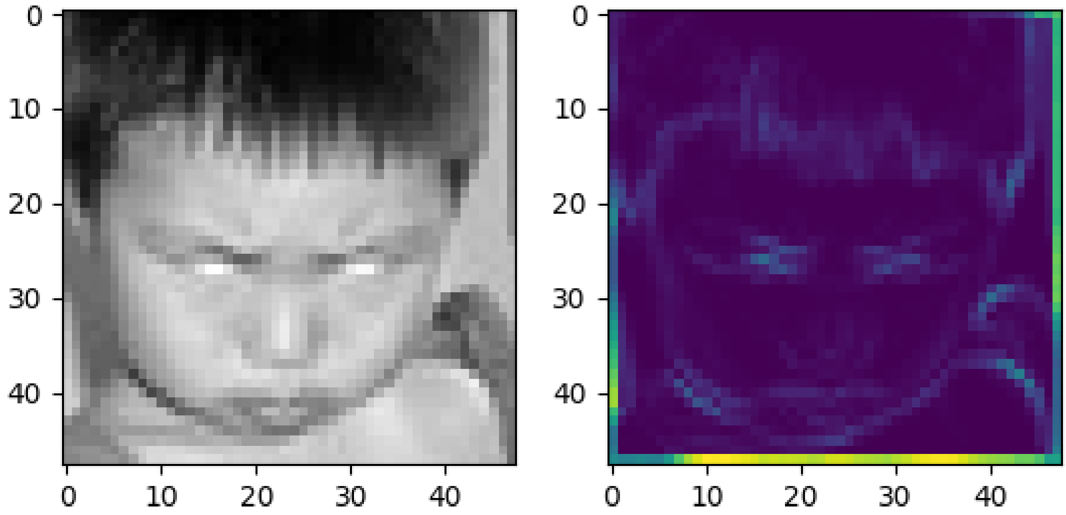


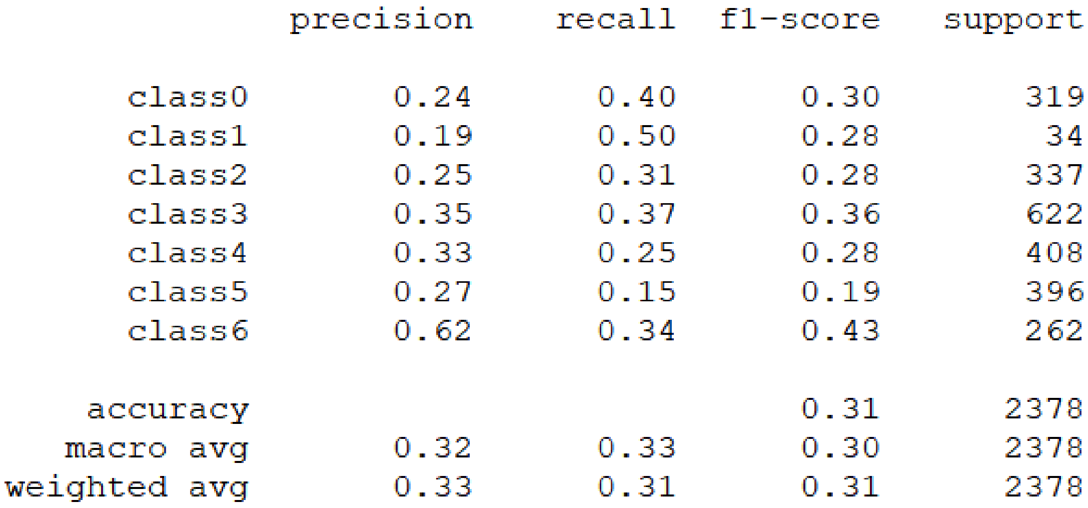
FIGURE 2 – LBP image feature extraction example.

After that, we will flatten the feature map(48,48) generated by LBP into a vector of 2304(48*48) which is the representation of corresponding image. So we can get a array of (28709, 2304) for all images. Notably, we found that there are some exceptions with value of NAN after applying LBP. We need to delete the abnormal data, and then we get (23774, 2304) finally.

2.3 Expression Recognition with KNN

(K-nearest neighbor, kNN) is a basic classification and regression method, and is a common method in supervised learning methods. The k-nearest neighbor algorithm assumes that given a training data set, the instance category has been determined. When classifying, for a new instance, prediction is made by means of majority voting and other methods according to the training instance category of its k nearest neighbors.

During this training process, we maintained a ratio of 0.9 for supervised learning, a ratio of 0.1 for testing and 3 neighbors for KNN. The classification report is shown as the flowing Figure 3



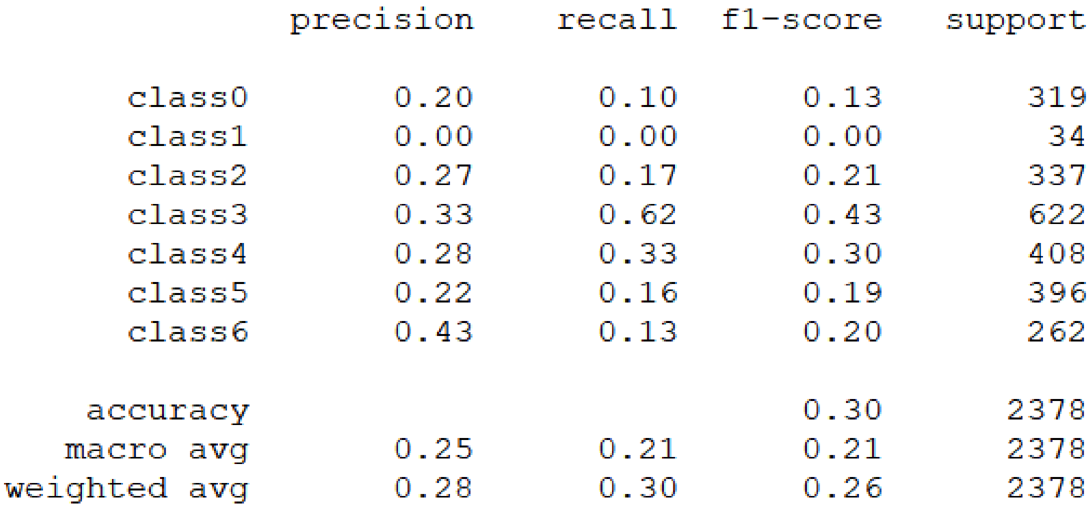
```
              precision    recall  f1-score   support

      class0       0.24      0.40      0.30       319
      class1       0.19      0.50      0.28        34
      class2       0.25      0.31      0.28       337
      class3       0.35      0.37      0.36       622
      class4       0.33      0.25      0.28       408
      class5       0.27      0.15      0.19       396
      class6       0.62      0.34      0.43       262

    accuracy                           0.31      2378
   macro avg       0.32      0.33      0.30      2378
weighted avg       0.33      0.31      0.31      2378
```

Figure 3 – KNN classification result report with 3 neighbors and a ratio of 0.9 for training.

In addition, we also try different neighbors(30) and different ratio for training. The results are shown as the following Figure 4 and Figure 5.

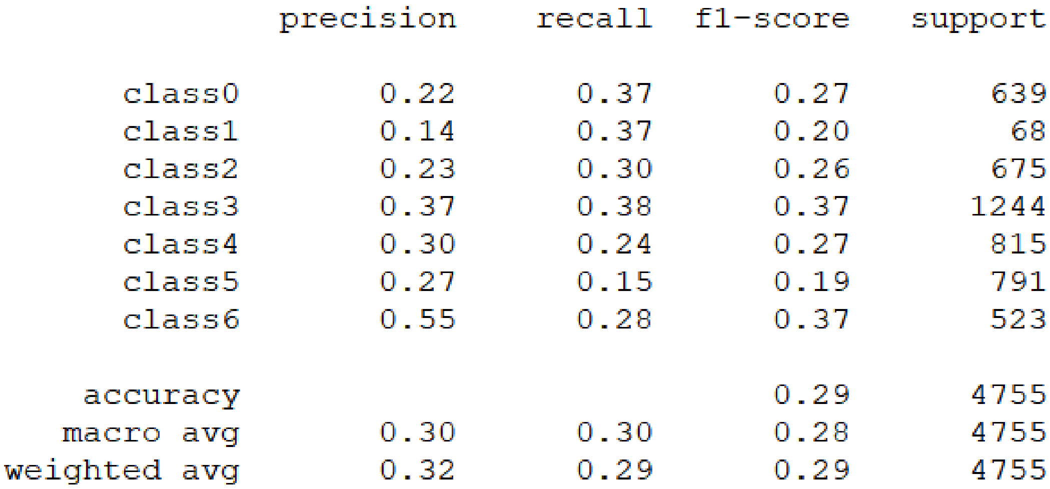


```
              precision    recall  f1-score   support

      class0       0.20      0.10      0.13       319
      class1       0.00      0.00      0.00        34
      class2       0.27      0.17      0.21       337
      class3       0.33      0.62      0.43       622
      class4       0.28      0.33      0.30       408
      class5       0.22      0.16      0.19       396
      class6       0.43      0.13      0.20       262

    accuracy                           0.30      2378
   macro avg       0.25      0.21      0.21      2378
weighted avg       0.28      0.30      0.26      2378
```

Figure 4 – KNN classification result report with 30 neighbors and a ratio of 0.9 for training.



```
              precision    recall  f1-score   support

      class0       0.22      0.37      0.27       639
      class1       0.14      0.37      0.20        68
      class2       0.23      0.30      0.26       675
      class3       0.37      0.38      0.37      1244
      class4       0.30      0.24      0.27       815
      class5       0.27      0.15      0.19       791
      class6       0.55      0.28      0.37       523

    accuracy                           0.29      4755
   macro avg       0.30      0.30      0.28      4755
weighted avg       0.32      0.29      0.29      4755
```

Figure 5 – KNN classification result report with 3 neighbors and a ratio of 0.8 for training.

Therefore, we can draw the following conclusions. A relatively large proportion is used for training to improve the KNN classification effect. This is because more samples of supervised learning make the model prediction more accurate. In addition, the number of neighbors for KNN does not significantly improve the classification performance.

In general, LBP and KNN are not very effective on this database. Maybe we should use stronger feature extractors and classifiers like CNN or DCNN.