Facial expression recognition based on opencv and keras

Li Xinyue
LanZhou University, UAIS CAI-minions Resort
Lanzhou University Annual meeting paper

Abstract

With the rapid development of deep learning and machine learning, facial expression recognition is the most direct understanding way of human emotion, ushered in a new way of processing. From traditional recognition methods to recognition methods based on deep learning, facial expression recognition has developed by leaps and bounds. Automatic analysis of facial expressions is very important for computers to understand human emotional state. it is an important way to understand human emotions based on computer vision. it has important influence and is used in many fields, such as human-computer interaction data-driven animation and so on.

Introduction

This paper uses Keras and Opencv for facial expression recognition by constructing a convolution neural network model. The choice of Keras is due to its ease of use and compatibility with TensorFlow. The network structure is crucial for accuracy, so the paper experiments with different models, including traditional neural networks, Xception, and residual networks. To prevent overfitting, model parameters are adjusted through repeated experiments.

Challenge

Since the expression recognition methods based on deep learning are affected by many hyperparameters, the comparability of the current facial expression recognition methods is not strong, and it is necessary to evaluate different expression recognition methods on different simple baseline methods.

Methods

Preprocess Data

We choose the fer2013 data set, which has a total of 35887 images. The dataset(Fig.1) is divided into three categories, each of which is as follows: 28709 training sets (Training), 3859 verification sets (Val) and 3859 test sets (Test).

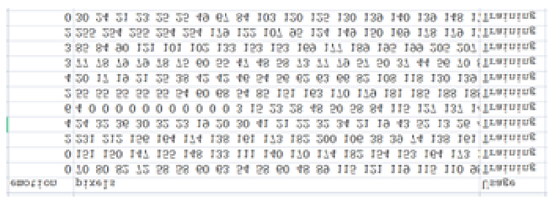

Figure 1: fer2013 dataset.

Data Enhancement

Data enhancement can be used to improve model performance by adding an appropriate amount of noise to the original data to improve model expressiveness and avoid overfitting. ImageDataGenerator can be used to perform data enhancement by generating an iterative object that specifies the desired enhancement operations[1], such as image decentralization, random flipping, and offsetting. After processing the images, they can be accessed through Next and saved in the Output file. The four-dimensional array received by Flow specifies the number, length, width, and grayscale of the image.

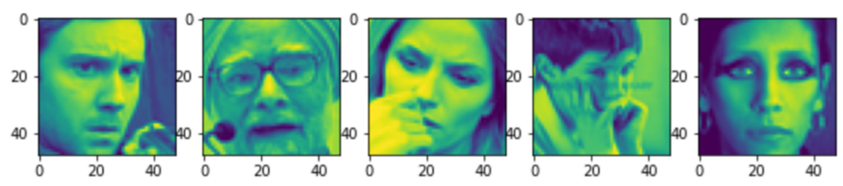

Figure 2: Visualization of original data.

By comparing the enhanced image (Fig.2) with the original image (Fig.3), the enhanced image works better.

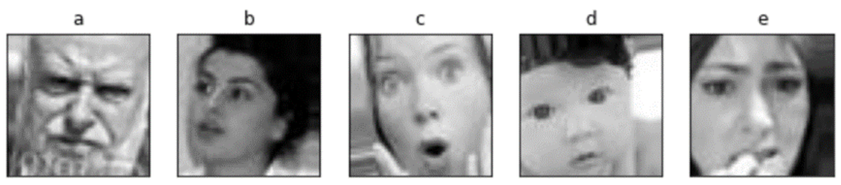

Figure 3: The data enhanced visualization.

Scale Normalization

To handle input images of varying sizes and ensure that key facial features are retained while irrelevant information is discarded, the images are cropped to a uniform size. However, the size of the crop should be appropriate - if it is too large, extraneous information will be included, making feature extraction more difficult and reducing model accuracy. Conversely, if the crop is too small, important information may be lost. Therefore, the human eye is typically used as a reference point for cropping and aligning the images.

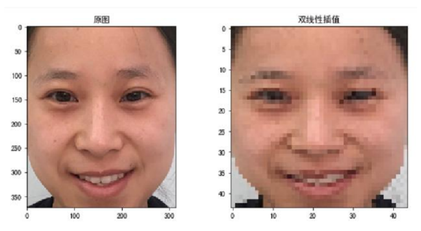

Figure 4: Compared the double-valued interpolation method's result with the original figure.

Model

Three models were compared in this experiment: Xception, traditional neural network, and residual network model[2]. The Xception model takes single-channel grayscale images with a shape of [48 × 48 × 1], and the traditional neural network and residual network models use multiple layers for feature extraction. To reduce memory cost and training time, the training data is randomly divided into batches. During training, each batch is fed into the model to calculate the predicted value, and the optimizer updates the model parameters to minimize the loss function based on the derivative of the loss function compared with the actual value. The training process saves optimal weight data using Keras callback mechanisms and monitors loss and accuracy to prevent overfitting or stagnation. A callback function is set to dynamically adjust the learning rate when the standard evaluation does not improve, and the training is stopped if the loss on the verification set remains the same after multiple adjustments.

Results

This chapter conducts an experiment using the fer2013 dataset with image size of [48 × 48]. The Xception, traditional, and residual network models are used for comparison, with a batch size of 32 and a total of 35887 images. Due to unstable computing resources, hierarchical training (50 or 100) is used, and GPU of the Googlecolaboratory is used for cloud training. The weight data is always in memory and video memory, and the training results can be reused in time. See Table1 for the comparison results.

Table 1: Model training comparison table

Model	Image size	Batch size	epochs	Train_acc	optimizer	Val_acc
Xception	48×48	32	200	0.72	adam	0.63
Tradition	48×48	32	200	0.68	adam	0.61
Resnet	48×48	32	200	0.81	adam	0.66

The traditional model achieved a training set accuracy of 0.68 and a verification set accuracy of 0.61 with over 300,000 parameters(Fig.6). On the other hand, the Xception model achieved a training set accuracy close to 0.72, a verification set accuracy of about 0.63, and had only 50,000 parameters, leading to a better performance compared to the traditional model.

```
Epoch 98/100
897/897 [==============================] - 13s 14ms/step - loss: 0.8370 - accuracy: 0.6848 - val_loss: 1.1158 - val_accuracy: 0.6046
Epoch 99/100
897/897 [==============================] - 13s 14ms/step - loss: 0.8376 - accuracy: 0.6868 - val_loss: 1.0875 - val_accuracy: 0.6208
Epoch 100/100
897/897 [==============================] - 12s 14ms/step - loss: 0.8329 - accuracy: 0.6859 - val_loss: 1.0914 - val_accuracy: 0.6140
```

Figure 5: The display of traditional model training results

```
=================================
Total params: 318,407
Trainable params: 318,407
Non-trainable params: 0
_________________________________
```

Figure 6: The parameters used in traditional models.

By selecting an image and selecting an image in the file for facial expression recognition, then you can see that the recognition result is anger as shown in Fig.7 below. The probability of anger recognition in the column on the right is 95%, and the rest of the disgust is slightly higher by 4%

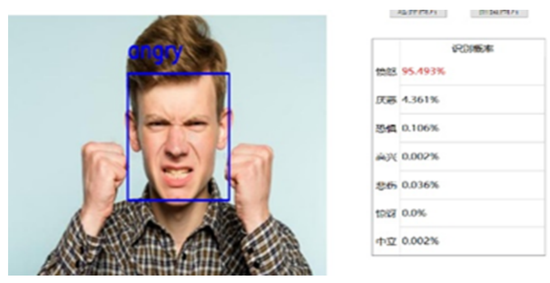

Figure 7: The display of facial expression recognition results.

Summary

This paper presents a comprehensive empirical study of facial expressions, which involves addressing various challenges such as variations in image size and quality, low-resolution images, and model selection and parameter optimization to improve recognition accuracy. These issues are critical for achieving accurate facial expression recognition, as different models can yield varying results on the same dataset.

References

[1] Hong Guo,Jiayou Chen. Dynamic Facial Expression Recognition Based on ResNet and LSTM[C]//Proceedings of 2019 2nd International Conference on Communication,network and Artificial Intelligence(cnai 2019): Iop Publishing, 2019: 990-996.

[2] Zhe Sun,Hehao Zhang,Suwei Ma, et al. Combining filtered dictionary representation based deep subspace filter learning with a discriminative classification criterion for facial expression recognition[J]. Artificial Intelligence Review, 2022(prepublish).