Facial Emotion Detection With CNN

Zhekai Liu
Final Project Summary
Stats-507 Data Science and Analytics using Python
zhekail@umich.edu

***Abstract*—This facial emotion detection project uses the CNN model in the dataset fer-2013. Work including data preparation, model structure, fine-tuning, and analysis with graph and labels.**

I. Introduction

Rapid advances in artificial intelligence have sparked interest in understanding how machines interpret and respond to human emotions, especially through facial expressions. Facial expressions play a crucial role in human communication by conveying emotions effectively and quickly. Deep learning approaches in human and computer interactions are used in artificial intelligence research as an effective system application process. This study proposes the development of a system that can predict and recognize the classification of facial emotions using the Convolution Neural Network (CNN) algorithm and feature extraction. The aim of this project is to explore how modern machine learning models, in particular Convolutional Neural Networks (CNNs), can recognize and classify human emotions based on facial expressions [1].

The goal of this project is to implement a CNN model for facial expression recognition using the FER-2013 dataset. By using CNN, this project focus on analyze the effectiveness of widely used machine learning techniques in recognizing emotional states and tries to optimize the network structure and the training process for better performance.

Facial expression recognition (FER) has seen significant advancements since its inception in the 1970s, especially with the rise of computational power in the 21st century. With many researchers and engineers working on fer-2013 dataset, current models achieve more than 0.75 accuracy with model ResEmoteNet and extra data from fer-2013 benchmark. The popular ann latest deep learning approaches, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and Transformers, have made notable progress in improving emotion recognition by automatically learning high-level features from images and using transfer learning to address data limitations. Instead of developing entirely new architectures, the focus is now on optimizing existing models to enhance their efficiency and accuracy [2].

II. Method

A. Data Description

This project uses fer-2013 which is a popular and classical dataset in Facial Expression Recognition field. I download the csv file from Hugging Face. This dataset has totally 35,887 rows and 3 columns. The first column contains 2304 integrals each row which means a 48*48 pixels image. The second column is integrals from 0 to 6 which represent seven different expression. Here are the labels and their counts: 0:Angry(4953); 1:Disgust(547); 2:Fear(5121); 3:Happiness(8989); 4:Sad(6077); 5:Surprise(4002); 6:Neutral(6198). The third column is the train-test split in original dataset, but this project would not use them [3].

B. Data Pre-processing

The data preprocessing process starts with avoiding order bias by disrupting the data, thus improving the generalization ability of the model. Sentiment labels were then one-hot coded to accommodate multiple classification tasks and enable the model to output the probability of each type of sentiment. When processing the pixel data, the image pixel values were segmented and normalized to between 0 and 1 to speed up training and avoid the problem of vanishing or exploding gradients. Next, the data is reshaped into a fixed input shape (48*48) required by the model to ensure that the size and number of channels of the image match the requirements of the convolutional neural network. The dataset was divided into a training set (81%), a validation set (9%) and a test set (10%), where the training set was used to train the model, the validation set was used for tuning the parameter and observing overfitting, and the test set was used for the final evaluation of the model's performance. In addition, data augmentation is used to perform random transformations on the images during the training process, such as panning, flipping, and scaling, which effectively increases the diversity of the data, prevents model overfitting, and improves its generalization ability. Through these steps, data preprocessing provides adequate preparation for model training and ensures the quality and diversity of the input data.

C. Parameters and Fine-Tune

Due to arithmetic limitations, smaller model would be more appropriate for this project, although may lead to limited learning ability of the model, thus affecting the performance of the model. In order to try the best, adding fine-tune parts and compare different parameters' results would be a nice method.

III. Model and Results

A. Model Structure

This project uses tensorflow instead of Pytorch to build CNN model since there are multiple layers and would be easy to check structure and fine-tune with changing optimizer as well as parameters.

The model is a classical Convolutional Neural Network (CNN) for facial expression recognition, consisting of multiple convolutional and fully connected layers to extract and classify emotional features. The input is a 48x48 grayscale image, and low-level features are extracted using 32 3x3 filters in the first convolutional layer, followed by ReLU activation to mitigate the gradient vanishing problem. The second convolutional layer extracts more complex features with 64 filters, followed by BatchNormalization to normalize activation values and accelerate convergence, MaxPooling2D to reduce computational complexity, and Dropout (0.2) to prevent overfitting.

The third convolutional block uses 128 filters, followed by BatchNormalization, MaxPooling, and Dropout to continue stabilizing model training. The fourth block increases the filters to 256 and adds L2 regularization to further suppress overfitting, while the fifth block uses 512 filters for higher-order feature extraction, with BatchNormalization and MaxPooling to reduce computational load.

The multidimensional feature maps are flattened into vectors for fully connected layers. Two fully connected layers (256 and 128 neurons) map extracted features to the classification space, with ReLU activation, BatchNormalization, and Dropout to suppress overfitting and improve generalization. The output layer, with 7 neurons and Softmax activation, converts predictions into probability values.

During training, EarlyStopping monitors validation accuracy (val_accuracy) and stops if no improvement occurs over 5 epochs, reverting to the best weights. ModelCheckpoint saves the best performing model based on validation accuracy. The model is trained for 50 epochs with a batch size of 64, using validation data for evaluation to ensure optimal performance.

B. *Result*

This models' could reach around 65% test accuracy. The graphs and table below are the results of training and validation loss with accuracy, as well as the prediciton result in test set for each label.

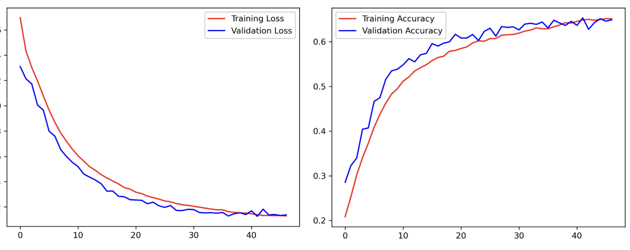

Fig. 1. Training and Validation Loss and Accuracy

The result plots show that both the Training Loss and Validation Loss are decreasing, which indicates that the model's learning process is effective and the loss is decreasing as training increases. The curve of decreasing loss is smooth, which means that the learning rate of the model is set appropriately and there is no excessive oscillation. The final validation accuracy and training accuracy are relatively

```
              precision    recall  f1-score   support

           0      0.616     0.528     0.568       489
           1      0.676     0.365     0.474        63
           2      0.650     0.381     0.481       522
           3      0.856     0.853     0.855       927
           4      0.516     0.581     0.547       597
           5      0.735     0.735     0.735       373
           6      0.512     0.714     0.596       618

    accuracy                          0.650      3589
   macro avg      0.652     0.594     0.608      3589
weighted avg      0.662     0.650     0.646      3589
```

close to each other, indicates that the model has a better generalization ability and does not overfit the training data. The validation accuracy stabilized around the 40th epoch and reached about 0.65, which indicates that the model converged better. However, the average performance of the model varied across categories, with some categories performing better than others. The reason is the number of "digest" label is much less than other labels. This class imbalance problem would cause model be more inclined to predict the majority of categories as a way to improve overall accuracy. In this way, a few categories (e.g., Disgust) may be ignored, causing the model to perform poorly on those categories. If the samples for a few categories are very small, the model may overfit these samples (i.e., the model remembers these samples instead of learning generalized patterns), resulting in poor performance in the test data.

In the fine-tune part, with multiple tries of different optimizer, the Adam optimizer perform best. For training epochs and learning rate, after set more epochs and lower learning rate, the model did not provide obviouly increase.

IV. Conclusion

Although the results' accuracy is less than the best methods on the same dataset, as a model which is not complex, easy for use and understanding, this result performs balance on between simplicity and efficiency after fine-tune. Future improvements may include addressing class imbalances, refining data augmentation techniques, or experimenting with different architectures and optimizers. Additionally, using larger, more balanced datasets could improve the model's ability to recognize emotions.

References

[1] P. Kaur, K. Krishan, S. K. Sharma, and T. Kanchan, "Facial-recognition algorithms: A literature review," *Medicine, Science and the Law*, vol. 60, no. 2, pp. 131–139, 2020.

[2] A. Younesi, M. Ansari, M. Fazli, A. Ejlali, M. Shafique, and J. Henkel, "A Comprehensive Survey of Convolutions in Deep Learning: Applications, Challenges, and Future Trends," *arXiv preprint*, arXiv:2402.15490, 2024.

[3] L. Zahara, P. Musa, E. P. Wibowo, I. Karim, and S. B. Musa, "The facial emotion recognition (FER-2013) dataset for prediction system of micro-expressions face using the convolutional neural network (CNN) algorithm based Raspberry Pi," in *Proc. 2020 Fifth Int. Conf. on Informatics and Computing (ICIC)*, Nov. 2020, pp. 1–9.