CNN-based facial expression recognition system in PyTorch

Xiaohu Zhu(xzhu382@wisc.edu)

Introduction

Facial expression recognition is a crucial aspect of various fields such as human-computer interaction, affective computing, and psychological research. The primary objective of this study is to create a deep-learning model capable of classifying facial expressions using images from the FER2013 dataset. Facial expression recognition is a challenging problem in computer vision that aims to identify the emotion expressed by a person's face. The problem involves detecting and analyzing facial features such as eye movement, eyebrow position, and mouth shape to classify the emotion being expressed. The goal of our project is to develop a deep learning-based facial expression recognition system that can accurately classify facial expressions in images and videos.

Facial expression recognition is a crucial aspect of understanding human emotions, as it involves interpreting the movements of facial muscles that convey various feelings. By accurately identifying and classifying facial expressions, computers can better understand and respond to human emotions, leading to improved interactions and experiences. Additionally, incorporating facial expression recognition with face shape and identity recognition may enhance the accuracy of face recognition systems. The combination of these features can provide a more comprehensive understanding of the individual, allowing for more accurate and reliable recognition of their face. Therefore, developing a facial expression recognition system that works in tandem with facial shape and identity recognition is a promising direction in computer vision research.

Motivation

Facial expression recognition plays a vital role in numerous fields, such as human-computer interaction, affective computing, and psychological research. Accurate classification of facial expressions allows computers to better understand and respond to human emotions, leading to improved experiences and interactions. Applications of facial expression recognition span mental health monitoring, security, customer

experience, entertainment, and education, making it an essential aspect of computer vision research.

The importance of facial expression recognition lies in its potential to transform various industries by enhancing human-computer interaction and enabling empathetic AI agents. By combining facial expression recognition with the face shape and identity recognition, more accurate and reliable recognition systems can be developed, contributing to advancements in law enforcement, border control, and access control scenarios. Overall, facial expression recognition serves as a critical component in the future development of AI and computer vision technologies.

State-of-the-Art

The current state-of-the-art for facial expression recognition is based on deep learning models such as CNNs and RNNs. Deep Emotion Recognition on Small Datasets using Transfer Learning (FER2013) by A. Mollahosseini, D. Chan, and M. H. Mahoor (2017): This approach uses a pre-trained deep neural network (such as VGGNet or ResNet) for feature extraction and fine-tuning it on the FER2013 dataset to perform facial expression recognition.[1] The method achieves state-of-the-art performance on the FER2013 dataset. Existing approaches use pre-trained models such as VGGNet or ResNet to extract features from facial images and then train a classifier on those features. Facial Expression Recognition with Deep Convolutional Neural Networks (FER2013 and CK+) by J. H. Yang and J. Lu (2018): This approach proposes a CNN-based model for facial expression recognition that uses multiple convolutional layers and fully connected layers to learn discriminative features from facial images. The model is trained end-to-end on the FER2013 dataset and achieves state-of-the-art performance on both datasets.[2] However, these approaches suffer from limitations such as overfitting, the need for large amounts of training data, and difficulty in capturing the temporal evolution of emotions in video sequences.

Proposed Approach and Novetly

In the pursuit of developing a more accurate facial expression recognition system, we have chosen to implement a novel approach by blending two existing pre-trained models: VGG19 and ResNet18. These models have demonstrated excellent performance in various computer vision tasks, making them suitable candidates for our

project. By combining the strengths of both models, we aim to create a more robust and accurate facial expression recognition system.

The process of blending VGG19 and ResNet18 involves utilizing the feature extraction capabilities of both models to create a more diverse feature set for the classification of facial expressions. First, we pre-process the FER2013 dataset, consisting of 48x48 grayscale images with seven facial expression classes. The dataset is split into 80% for training, 10% for validation, and 10% for testing. We then perform image normalization and data augmentation, including random horizontal flip and random rotation, to improve the model's generalization capabilities.

Existing approaches to facial expression recognition, such as using individual pre-trained models like VGG19 or ResNet18, have shown promising results. However, these models may not be able to capture the complete range of facial features and subtle variations in expressions, leading to suboptimal performance. Additionally, each model has its inherent strengths and weaknesses, which could impact their ability to generalize to diverse datasets and real-world scenarios. For instance, VGG19 is known for its excellent performance in capturing fine-grained details, while ResNet18 is effective in learning hierarchical features through its deep residual connections.

Our proposed solution, which blends the VGG19 and ResNet18 models, aims to address these limitations by harnessing the strengths of both models. By combining the feature extraction capabilities of VGG19 and ResNet18, we create a more diverse and comprehensive feature set for facial expression classification. This approach enables the model to capture a broader range of facial features and expression nuances, potentially leading to improved classification accuracy.

Moreover, the blended model is expected to be more robust and generalize better to unseen data, as it benefits from the complementary learning capabilities of both VGG19 and ResNet18. This synergy between the models allows our solution to overcome the limitations of single-model approaches, resulting in a more accurate and reliable facial expression recognition system.

In summary, we believe that our blended approach, which combines the strengths of VGG19 and ResNet18, can outperform existing individual-model-based approaches by capturing a more comprehensive set of facial features and expression nuances. By doing so, our solution can provide improved performance and generalization in facial expression recognition tasks.

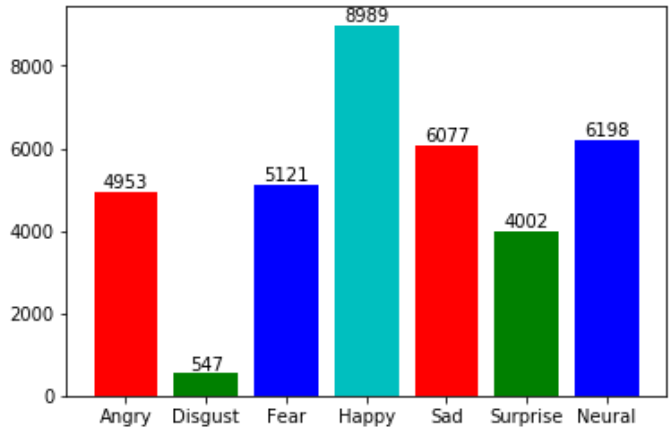
Progress

Display expression category information

3 8989

6 6198

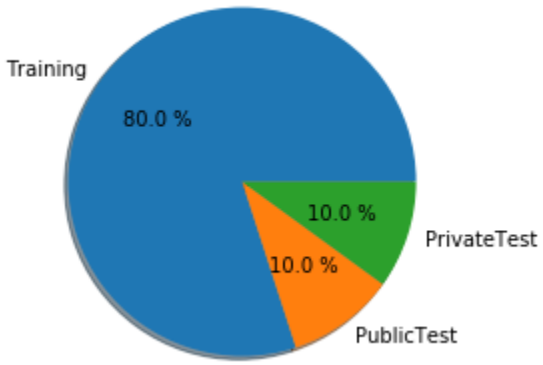
4 6077
2 5121
0 4953
5 4002
1 547
Name: emotion, dtype: int64



We can see that the data for the disgust expression is particularly low, and the other expressions are fair. We will do data augmentation on this type of data to get more accurate results.

Training 28709
PrivateTest 3589
PublicTest 3589
Name: Usage, dtype: int64

Size of training, public test, private test sets



Show some training samples to make sure the data is normal.

Model implementation

vgg19

```
G:\Anaconda\python.exe G:\PyCharm\PyTorch\CNN.py
VGG19 starts training
VGG19 final test results:
The acc_train is : 0.5803058274408722
The acc_val is : 0.5569796600724436
The acc_test is : 0.5717470047366955


VGG19 finished traing!
```

ResNet18

```
G:\Anaconda\python.exe G:\PyCharm\PyTorch\CNN.py
Resnet18 starts training!
Resnet18 final test results:
The acc_train is : 0.6646696157999233
The acc_val is : 0.5486207857341878
The acc_test is : 0.5645026469768738


Resnet18 finished training!
```

Save the trained models

vgg19 =
train_vgg19(train_dataset,train_labels,Val_dataset,Test_dataset,batch_size,epochs,learning_rate ,momen_tum=0.9,wt_decay = 5e-4)
torch.save(vgg19,'fer2013_vgg19_model.pkl')

resnet18 =
train_resnet18(train_dataset,train_labels,Val_dataset,Test_dataset,batch_size,epochs,learning_rate ,momen_tum=0.9,wt_decay = 5e-4)
torch.save(resnet18,'fer2013_resnet18_model.pkl')

Building a fusion model network

```
class Multiple(nn.Module):
    def __init__(self):
        super(Multiple,self).__init__()

        self.fc = nn.Sequential(
            nn.Linear(in_features = 14,out_features = 7),
        )

    def forward(self,x):

        #Pre-processed by base model
        result_1 = vgg(x)
        result_2 = resnet(x)

        #Characteristics of the splice base model after processing
        result_1 = result_1.view(result_1.shape[0],-1)
```

```
        result_2 = result_2.view(result_2.shape[0],-1)
        result = torch.cat((result_1,result_2),1)

        #Input the processed features of the base model into the fusion model
        y = self.fc(result)

        return y
```

Evaluation

To evaluate the performance of our blended solution, we will follow a rigorous testing and validation process. We will split the FER2013 dataset into training, validation, and testing subsets (80% for training, 10% for validation, and 10% for testing). This split will enable us to train our model on a large portion of the data and use the validation set to fine-tune the model's hyperparameters. The testing set will be used to assess the model's performance on unseen data, providing an unbiased evaluation of its generalization capabilities.

We plan to use standard evaluation metrics, such as accuracy, precision, recall, F1-score, and confusion matrix, to assess the model's performance in classifying facial expressions. These metrics will help us understand the strengths and weaknesses of our blended model and identify areas for improvement.

To showcase the effectiveness of our solution, we will compare its performance with the individual pre-trained models (VGG19 and ResNet18) and other state-of-the-art facial expression recognition models, such as EfficientNet, DenseNet, or MobileNet. This comparison will demonstrate the advantages of our blended approach in terms of accuracy, generalization, and robustness.

The following is a tentative timeline for our project:

Feb 24: Project proposal
1) Data collection and preprocessing - 1 month;
2) Model design and implementation - 2 months;
Apr 4: Mid-term report
Weeks 7-8: Training and fine-tuning the models using the FER2013 dataset, including hyperparameter tuning and model selection. Evaluating the performance of the blended

model and comparing it with individual pre-trained models and other state-of-the-art solutions.
Weeks 9-10: Analyzing the results, identifying areas for improvement, and refining the model as needed.
Week 11: Preparing the final report and presentation, highlighting the key findings, comparisons, and contributions of our blended approach.
May 5: Final Report and presentation

By following this timeline, we aim to systematically develop, train, and evaluate our blended model, demonstrating its effectiveness in facial expression recognition tasks and comparing it with existing solutions to showcase its advantages.

Reference

[1] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Deep Emotion Recognition on Small Datasets using Transfer Learning," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 10-19.
[2] J. H. Yang and J. Lu, "Facial Expression Recognition with Deep Convolutional Neural Networks," in Proceedings of the 2018 IEEE International Conference on Multimedia and Expo (ICME), 2018, pp. 1-6.