Report

Task:

Given the dataset, build a computer vision model to detect emotions from facial images.

Requirements for reproducing code:

- Have the packages installed like:
  - PyTorch
  - numpy
  - pandas
  - matplotlib
  - sklearn
  - timm
  - tqdm
  - albumentations
  - cv2
  - time
- Edit the root directory as mentioned in the comments in the code where your dataset is located
- For inference, mention the location for the models' .pth files according to the code and the comments beside it.

Approach and Understandings:

1. Understand the data
   a) Image size (fixed) = 48x48
   b) Image is single channel i.e., grayscale
   c) Generate dataframe based on file structure given
2. For training:
   a) Make a configuration for the training procedure with model names, size of image, number of epochs etc.
   b) Create the dataset class custom to the given dataset
   c) Define the augmentations to done to the images
   d) Define the model class and create the model using timm, also load the pre-trained weights of the said model
   e) Define the utility functions like loss function, LR scheduler, optimizer etc.
   f) Define the single epoch training function, looping over the train dataloader
   g) Define the single epoch validation function, looping over the validation dataloader
   h) Call all the functions in order
      1. Define dataset

2. Define dataloader
3. Loop over the models mentioned in the configuration
4. Create the model
5. Define the optimizer, scheduler and loss function
6. Loop over the number of epochs
7. Generate outputs from the model
8. Perform back-prop over the results and the ground truth labels
9. Evaluate the model over the validation set
10. Save the model of current validation score is better than previous validation score

3. For testing:
   a) Make a configuration for the testing procedure with model names used in training, size of images etc.
   b) Create the dataset class custom to the given dataset
   c) Define the augmentations to done to the images
   d) Define the model class and create the model using timm
   e) Define the utility functions like metric and ensemble.
   f) Define the single epoch testing function, looping over the test dataloader
   g) Call all the functions in order
      1. Define dataset
      2. Define dataloader
      3. Loop over the models mentioned in the configuration
      4. Create the model
      5. Generate outputs from the model
      6. Evaluate the model over the test set
   h) Create an ensemble of the outputs of the three models
   i) Evaluate the ensemble over the test set

Methodology:

The methodology is quite simple and explained with well in the essence of the approach:

- Keep everything modular and ensure individual parts work well within themselves
- Modularity ensures easy bug fixing and error handling
- Integrate the individual parts in logical fashion
- Check whether the outputs and inputs to these modular units are correct and can be accepted

Best scores of all models evaluated (on validation set):

se-resnet101 = 26%
resnet101 = 57%
resnet152 = 55%
wide-resnet50 = 66.5% (selected for ensemble)

efficientnet_b0 = 61%
mobilenetv3_large = 59%
regnetx_160 = 67% (selected for ensemble)
regnetx_320 = 68.2% (selected for ensemble)

Thank You