All things to know Emoji Project.

Facial expressions used:

0 - Neutral
1 - Smile/Happy
2 - Sad
3 - Wink
4 - Kiss
5 - Surprised
6 - Angry
7 - Monkey face
8 - Wink with tongue out
9 - Scared/Terrified
10 - Disgusted

How to use this project:

Create your facial expression dataset:

1. Start this file

```
python create_dataset_webcam.py
```

2. It will ask for label i.e. facial expression id (more about it later), number of pictures you want to take, and starting image number. You can take as many pictures for each expression as you want but make sure you do it using different lighting conditions, facial poisitions etc. Also make sure you take same number of images for each gesture or else you might introduce a bias. I usually keep it to 250.
3. For the starting image number, make sure you check the images in the dataset/ folder. If the file name of the last file is 249.jpg then you should enter 250 i.e. (last image number + 1)
4. The images will be stored in the new_dataset/ folder.

Retraining with the new_dataset (HARDER way):

You will see why this method is a bit hard.

Load the images of the new_dataset/

1. Start the load_images.py file

```
python load_images.py
```

2. Here you will be asked for which dataset folder to use. Enter 'new_dataset/'
3. The images will be stored as pickle file.
4. You will get 6 pickle files viz train_images, train_labels, test_images, test_labels, val_images, and val_labels

Retrain the model

1. Start the retrain_cnn_keras.py file.

   ```
   python retrain_cnn_keras.py
   ```

2. Here you will be asked for the trained model file name, new model file name, learning rate, epochs, and batch number.
3. For the trained model file name enter cnn_model_keras.h5
4. For the new model file name you can enter anything. A warning though, if you keep it blank or enter cnn_model_keras.h5, it will replace the original model when training if the validation accuracy increases from the previous step.
5. What I usually do is I enter cnn_model_keras1.h5 or something like that so that I do not mess it up.
6. For the rest of the hyper parameters, they will depend on how large the new_dataset/ is.
7. If the number of images for each expression is >= 250, I usually keep the default learning rate, 10-20 epochs and a batch size of 100.
8. After the training you will see the accuracy of the model.

Check the model's accuracy against the dataset/ folder

1. Start the load_images.py file

   ```
   python load_images.py
   ```

2. Here you will be asked for which dataset folder to use. Enter 'dataset/'
3. Start the compute_accuracy.py file

   ```
   python compute_accuracy.py
   ```

4. You will be asked to enter the model name. Enter the new model name. Usually in my case it is cnn_model_keras1.h5.
5. If the accuracy is small i.e the CNN error is greater than 4-5% try retraining the model again. That is, do the 2nd step 'Retrain the model'. Only now you need to use cnn_model_keras1.h5 as the model file name and you can keep the new model file name as blank or anything you want.
6. The lower accuracy happens perhaps due to the incorrect settings of the hyperparameters. Please let me know if I am doing it wrong. Need some expert advice here.

Training the model from the beginning (EASIER way)

1. After you have stored your images in the new_dataset/ folder, move the contents of the folder to the dataset folder.

2. Start the train_cnn_keras.py file

```
python train_cnn_keras.py
```

Start emojify

1. If you are satisfied with the accuracy of the model then you can start the emoji.py file.

   ```
   python emojify.py
   ```